https://speakerdeck.com/rmoff/

Processing IoT data with Apache Kafka, KSQL, and Machine Learning

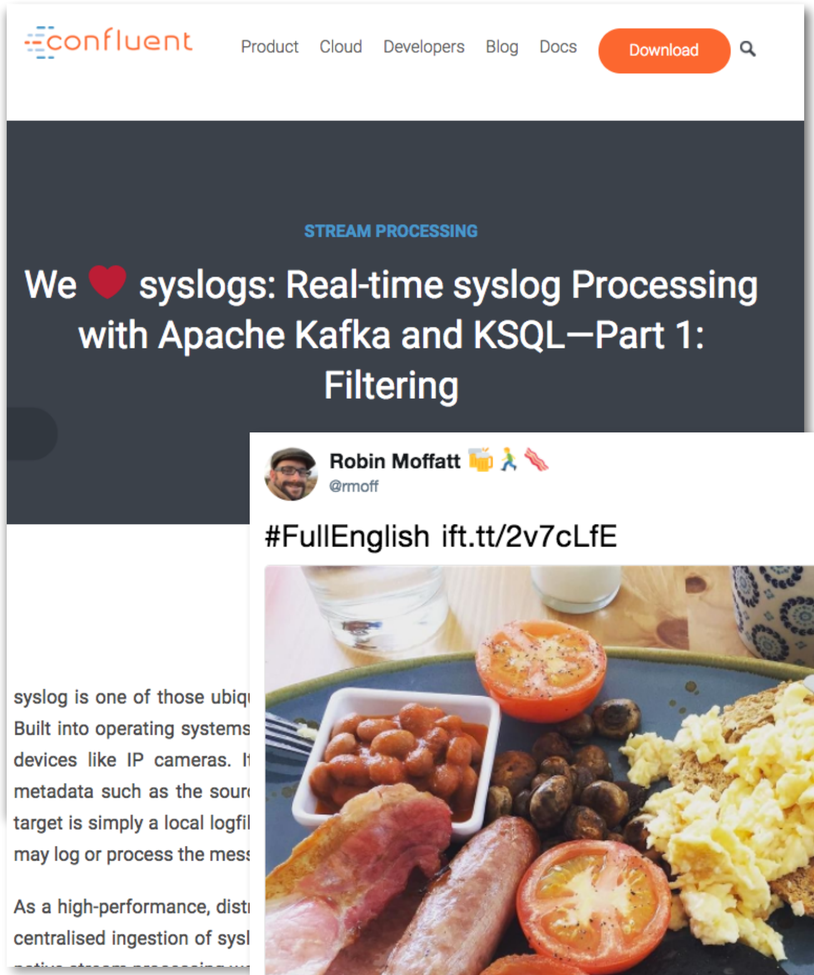
building IoT, Köln

5th June 2018 / Robin Moffatt

@rmoff robin@confluent.io

$ whoami

- Developer Advocate @ Confluent
- Working in data & analytics since 2001
- Oracle ACE Director & Dev Champion
- Blogging : http://rmoff.net & http://cnfl.io/rmoff
- Twitter: @rmoff
  - Geek stuff
  - Beer & Fried Breakfasts



https://speakerdeck.com/rmoff/

What we will talk about…

- What is Apache Kafka?
- IoT and Kafka
- Machine Learning and Kafka
- Stream Processing with Kafka
- Put it all together :

  IoT + Stream Processing + ML + Kafka!

What we will talk about…

- **What is Apache Kafka?**
- IoT and Kafka
- Machine Learning and Kafka
- Stream Processing with Kafka
- Put it all together :

  IoT + Stream Processing + ML + Kafka!

“what is kafka?

“Kafka is a streaming platform

“all your data is event streams

A Sensor Reading

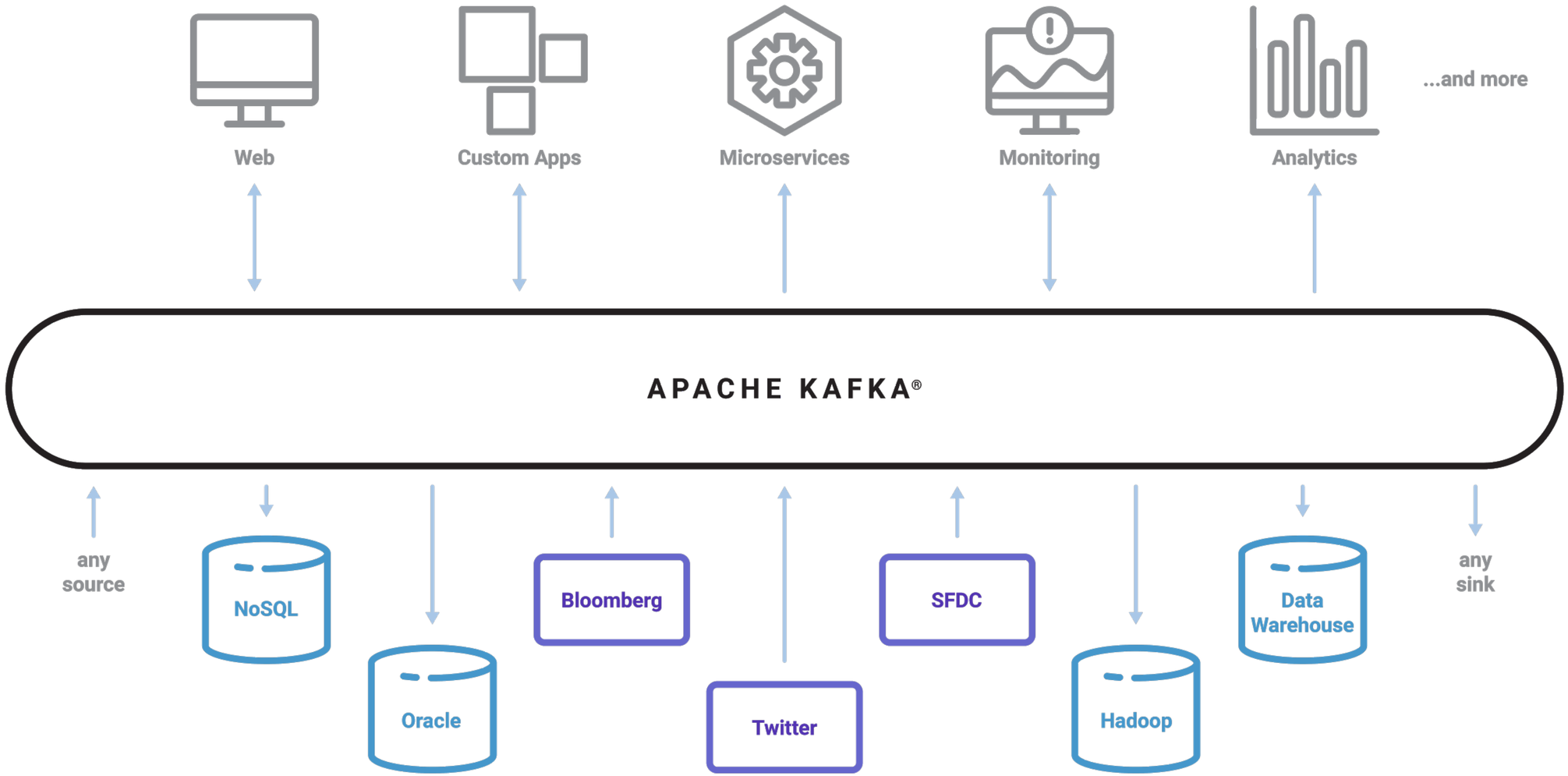
An Application Log Entry

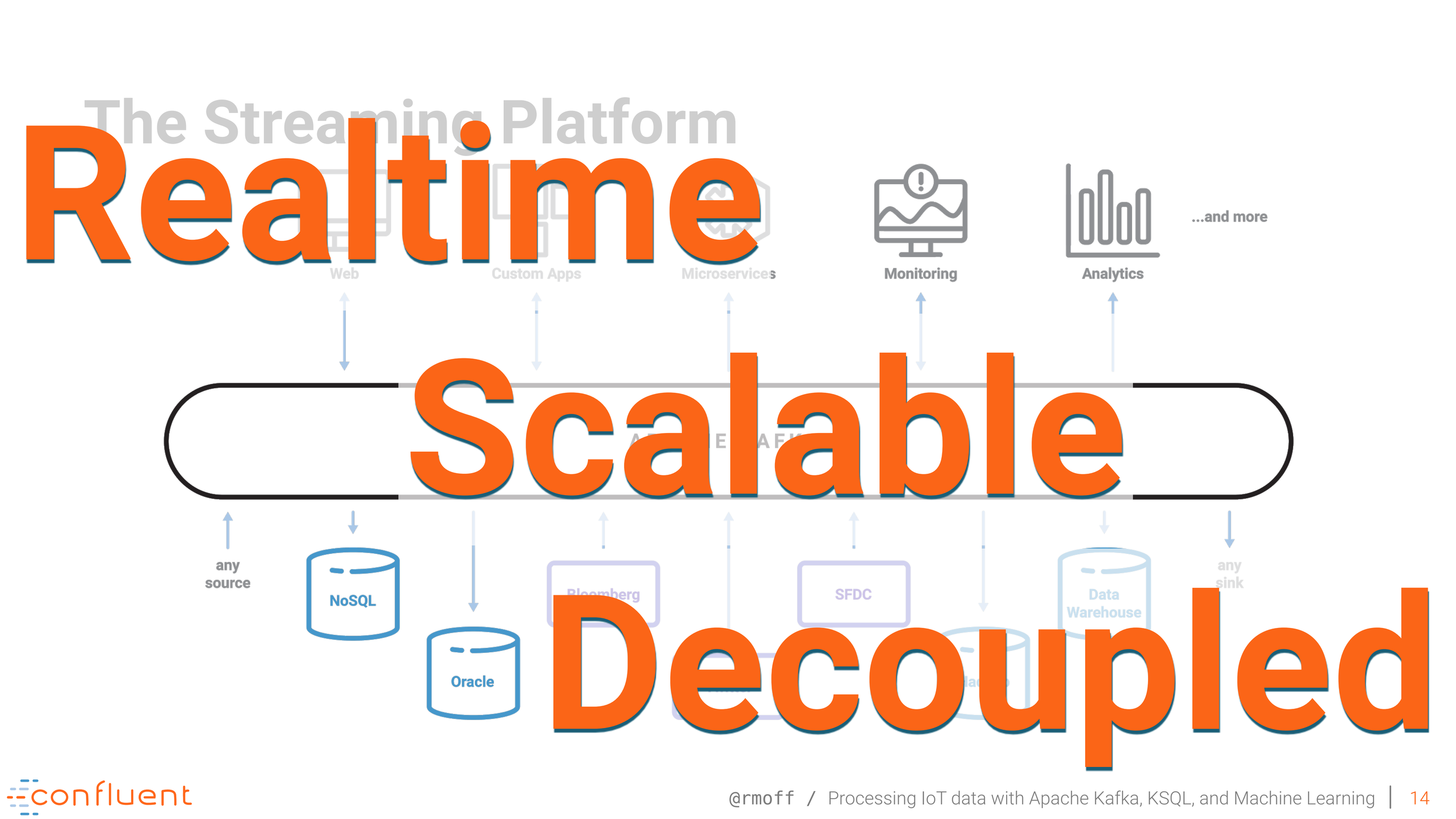
A Customer Experience

A Sale

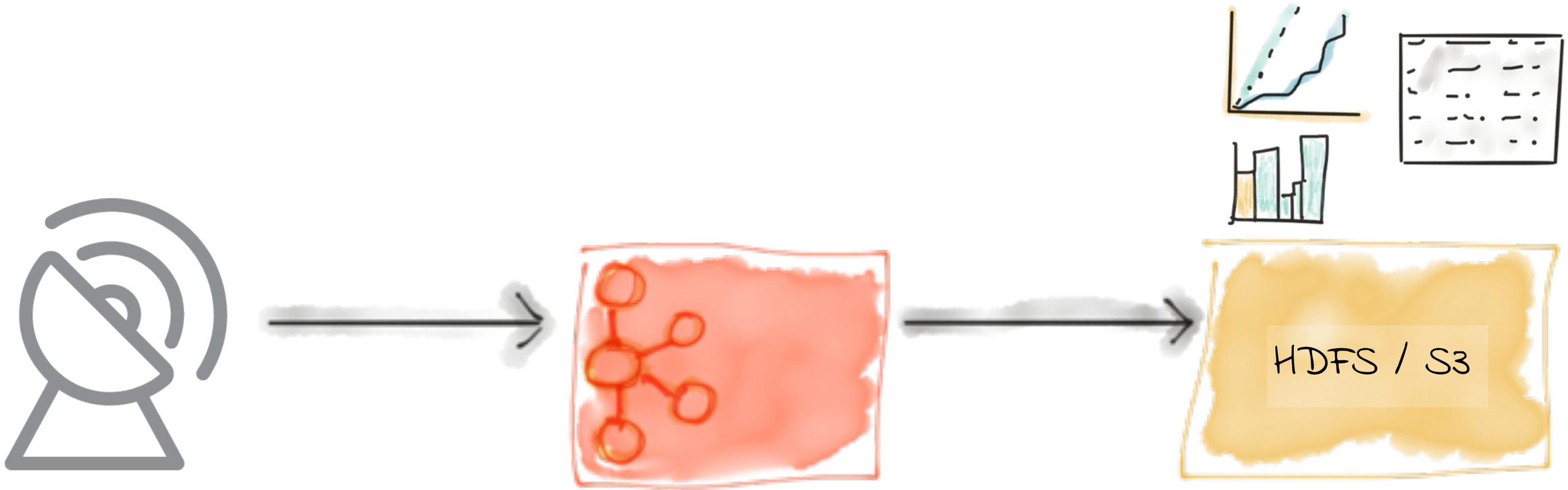
Databases

The Streaming Platform

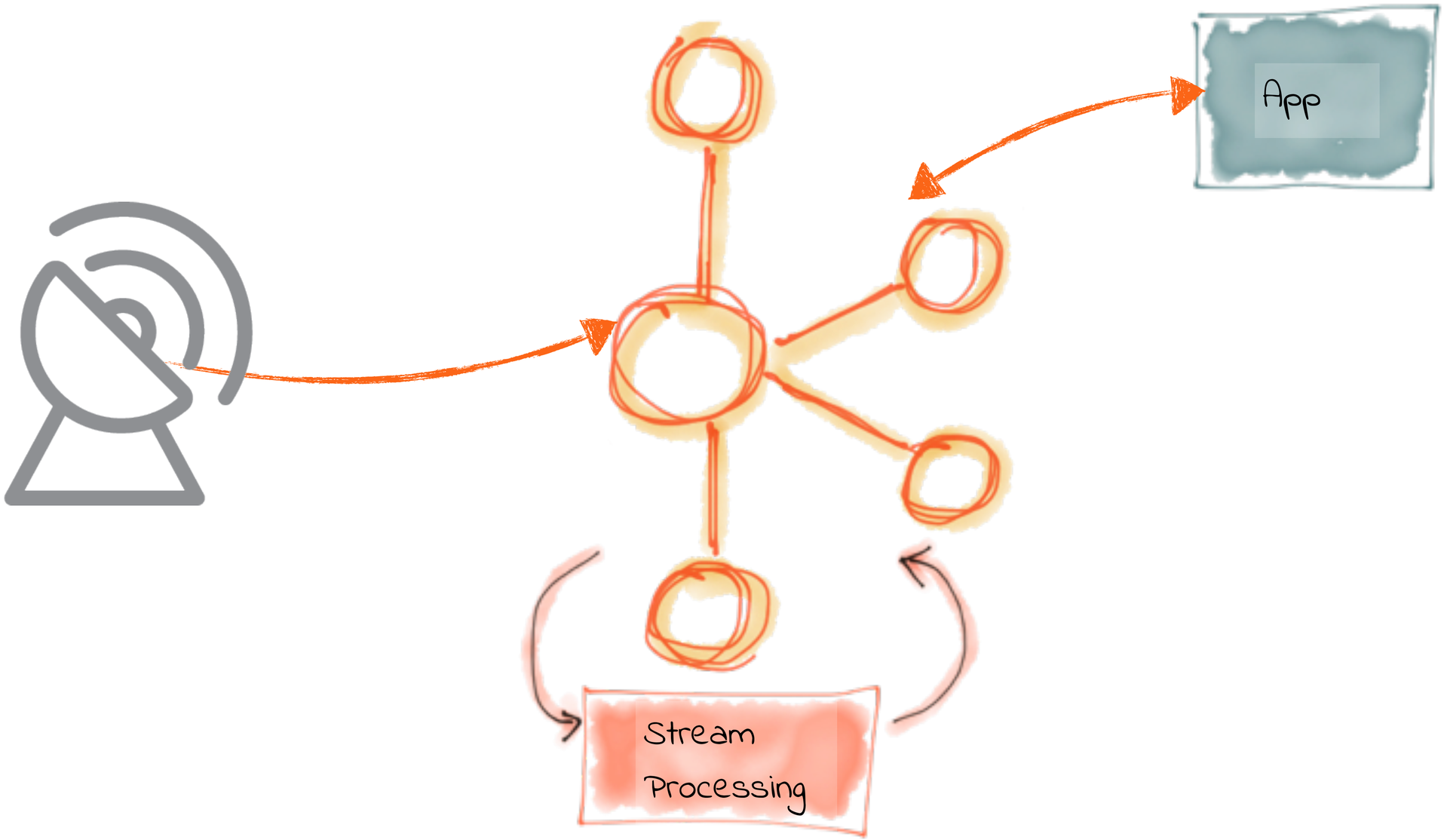



The Streaming Platform
Realtime
Scalable
Decoupled
Web
Custom Apps
Microservices
Monitoring
Analytics
...and more
any source
NoSQL
Oracle
Bloomberg
SFDC
Data Warehouse
any sink

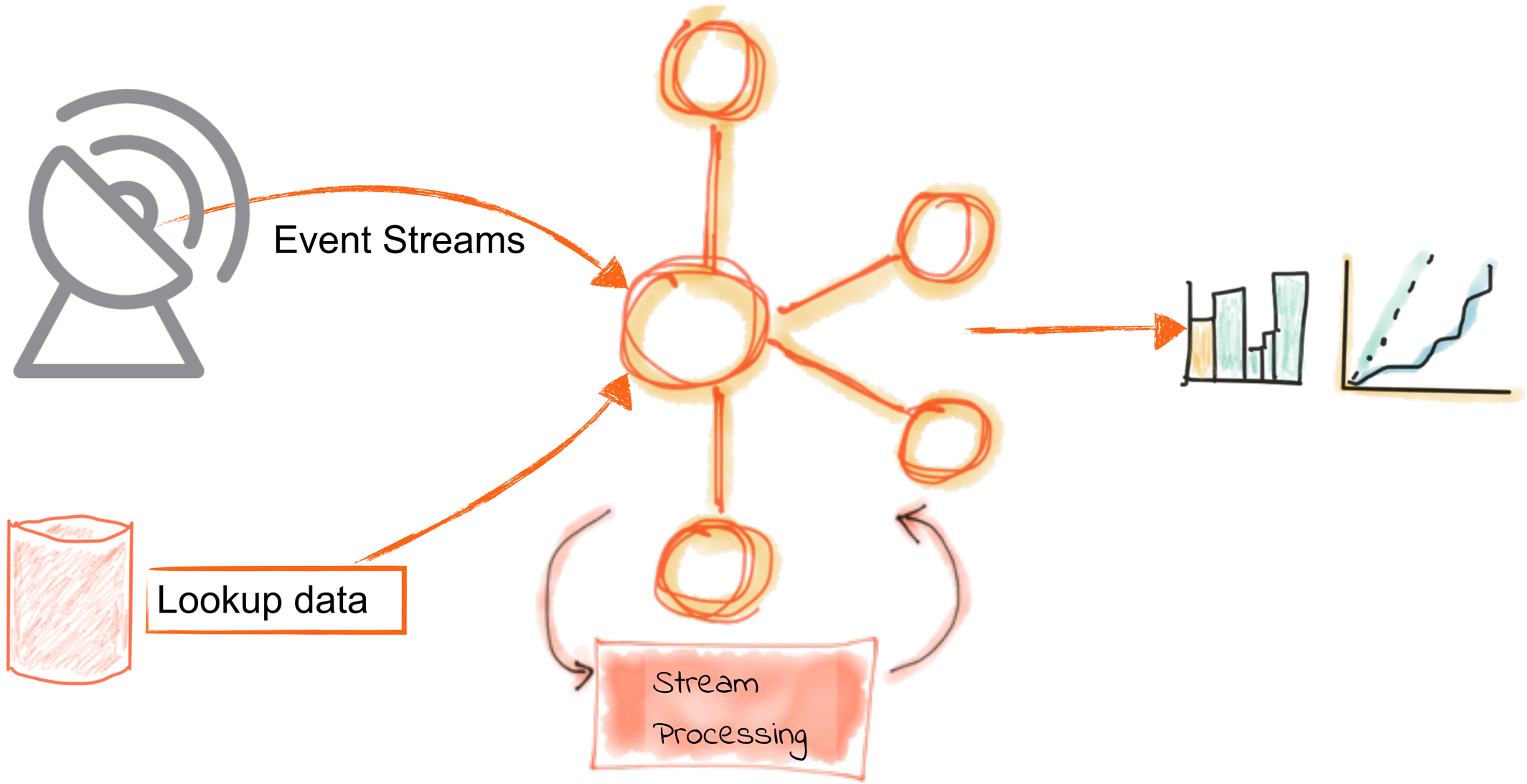
Simple Data Pipelines - storage in HDFS for batch analytics / processing



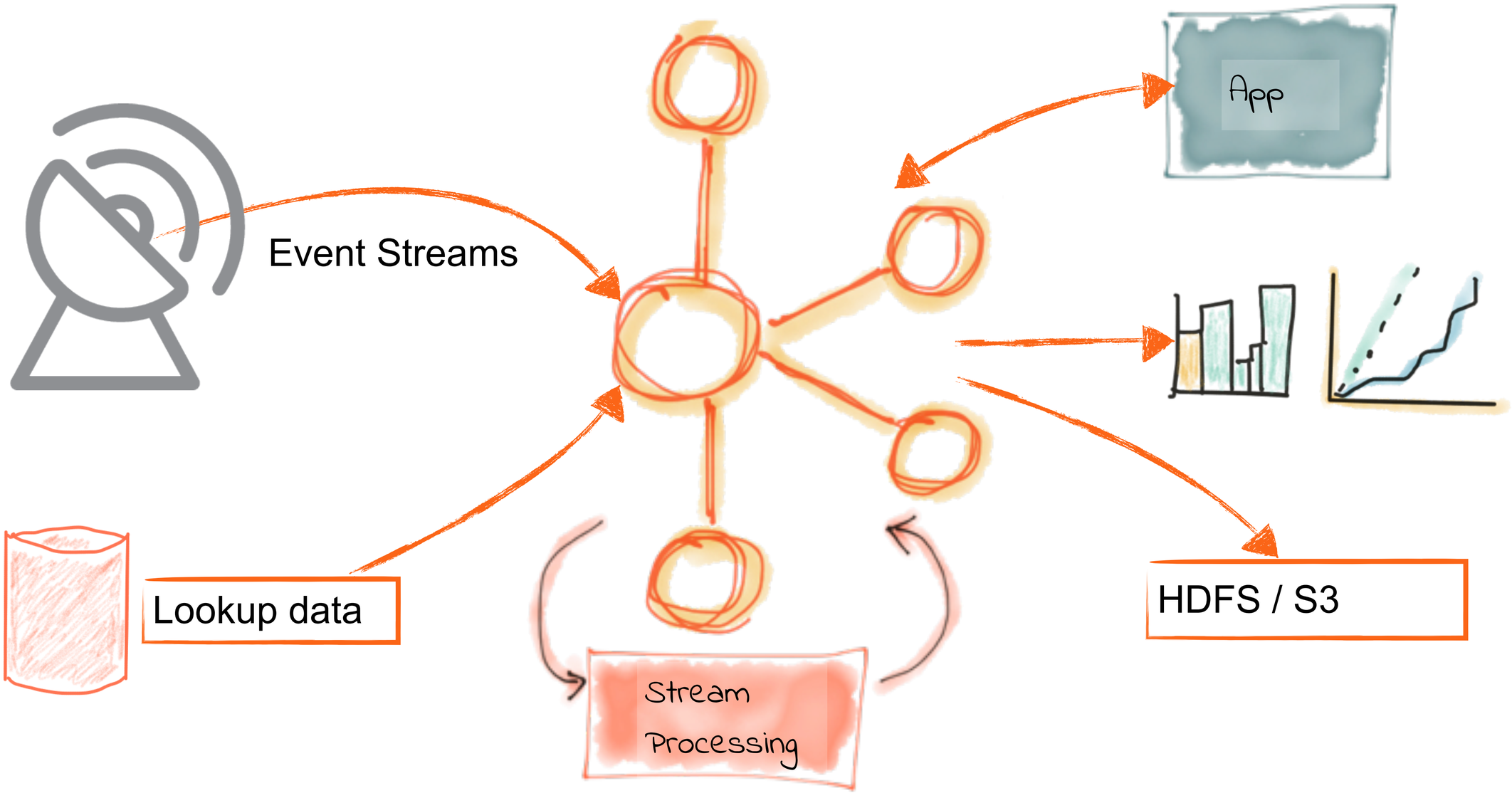
Event Driven Applications



Real-time Enrichment and Analytics on What's Happening *Now*



Real-time Enrichment and Analytics on What's Happening *Now*



Apache Kafka at Large Scale

LinkedIn (2018)

Operation Challenges

- The scale of Kafka deployment @LinkedIn
  - 2,100+ brokers
  - ~ 60,000 topics
  - ~ 4.2 million partitions
  - > 4.5 trillion messages / day
- Huge operation overhead
  - Hardware failures are norm
  - Workload skews

Strata DATA CONFERENCE

NETFLIX (2018)

Kafka @ Netflix Scale

- 4,000+ brokers and ~50 clusters in 3 AWS regions
- > 1 Trillion messages per day
- At peak (New Years Day 2018)
  - 2.2 trillion messages (1.3 trillion unique)
  - 6 Petabytes

QCon

https://conferences.oreilly.com/strata/strata-ca/public/schedule/detail/63921
https://qconlondon.com/london2018/presentation/cloud-native-and-scalable-kafka-architecture

Apache Kafka at Large Scale

LinkedIn (2018)

Operation Challenges

- The scale of Kafka deployment @LinkedIn
  - 2,100+ brokers
  - ~ 60,000 topics
  - Hardware failures are norm
  - Workload skews

> 4.5 trillion messages / day

Strata DATA CONFERENCE

NETFLIX (2018)

Kafka @ Netflix Scale

- 4,000+ brokers and ~50 clusters in 3 AWS regions
- > 1 Trillion messages per day
- At peak (New Years Day 2018)
  - 2.2 trillion messages (1.3 trillion unique)
  - **6 Petabytes**

QCon

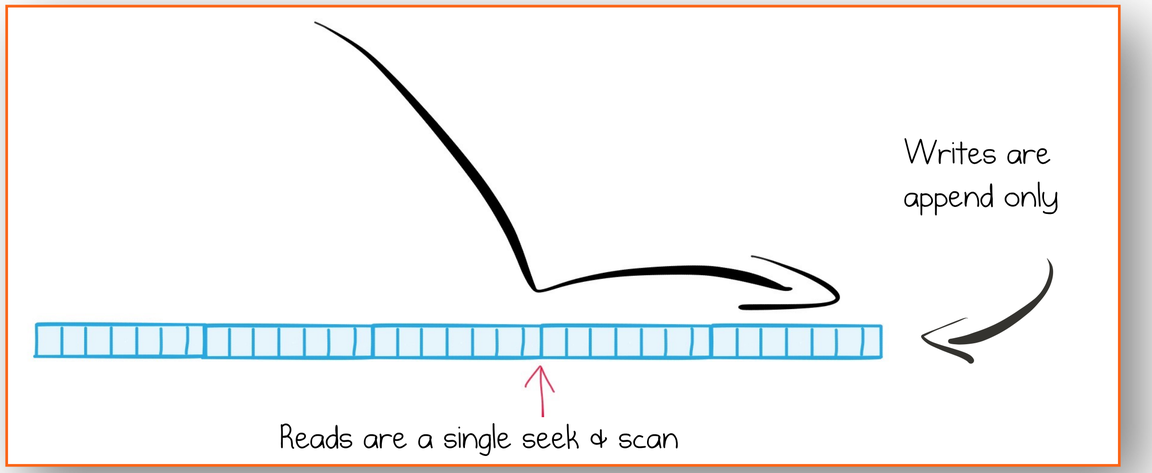
https://conferences.oreilly.com/strata/strata-ca/public/schedule/detail/63921
https://qconlondon.com/london2018/presentation/cloud-native-and-scalable-kafka-architecture

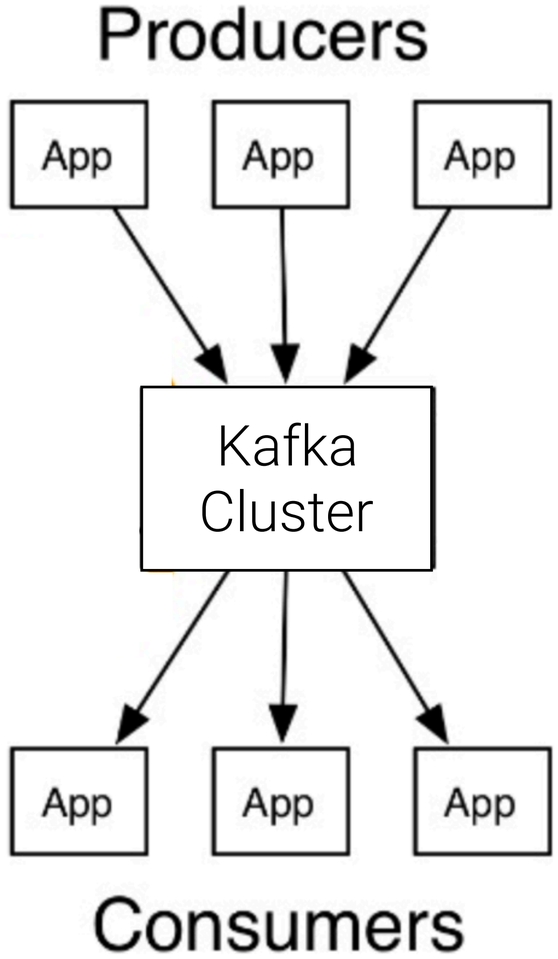
What we will talk about…

- What is Apache Kafka?
- **IoT and Kafka**
- Machine Learning and Kafka
- Stream Processing with Kafka
- Put it all together :
  IoT + Stream Processing + ML + Kafka!

Kafka

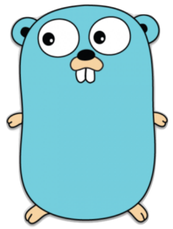
A **Distributed Commit Log**. Publish and subscribe to streams of records. Highly scalable, high throughput. Supports transactions. Persisted data. Stream processing.





Producer and Consumer API

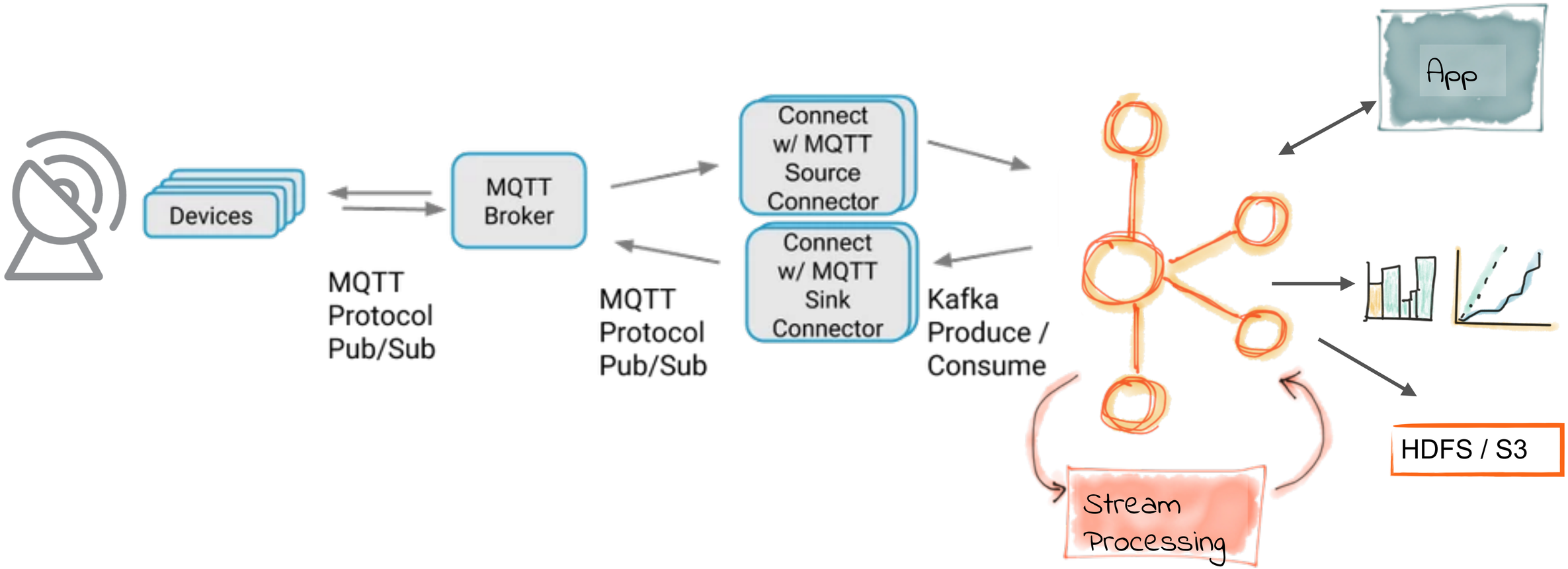
- Send data directly to Kafka using the Producer API
- Configurable for number of acks required etc
- Open-source client libraries available for:
  - Java
  - Scala
  - Python
  - C / C++
  - .NET
  - Go



Go

MQTT

- Very lightweight messaging protocol, often used with IoT devices
- Integrates with Kafka through **Kafka Connect**
  - https://github.com/jcustenborder/kafka-connect-mqtt



The Connect API of Apache Kafka®

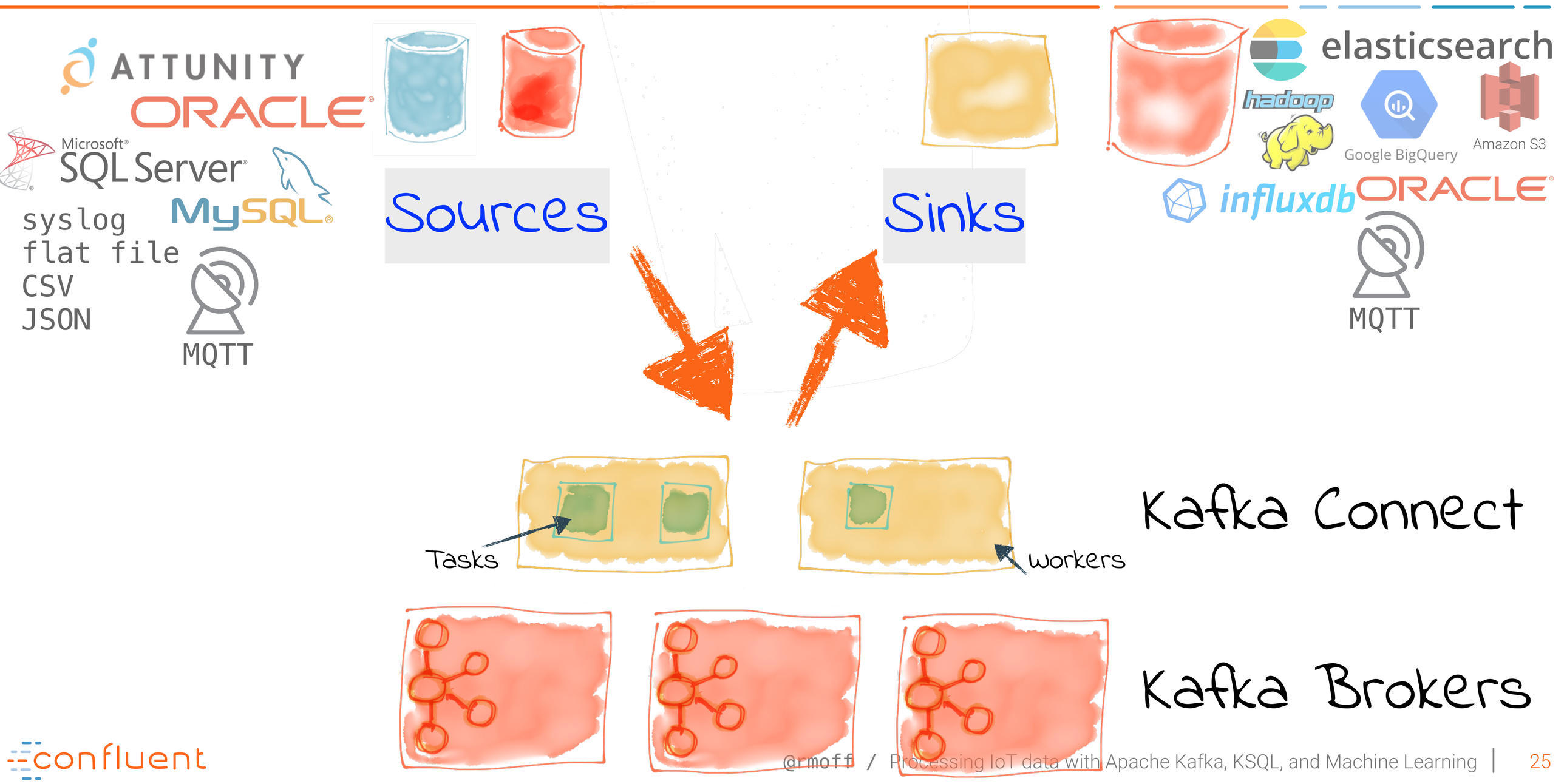
Reliable and scalable integration of Kafka with other systems – no coding required.

- ✓ Centralized management and configuration
- ✓ Support for hundreds of technologies including RDBMS, Elasticsearch, HDFS, S3
- ✓ Supports CDC ingest of events from RDBMS
- ✓ Preserves data schema
- ✓ Fault tolerant and automatically load balanced
- ✓ Extensible API
- ✓ Single Message Transforms
- ✓ Part of Apache Kafka, included in Confluent Open Source

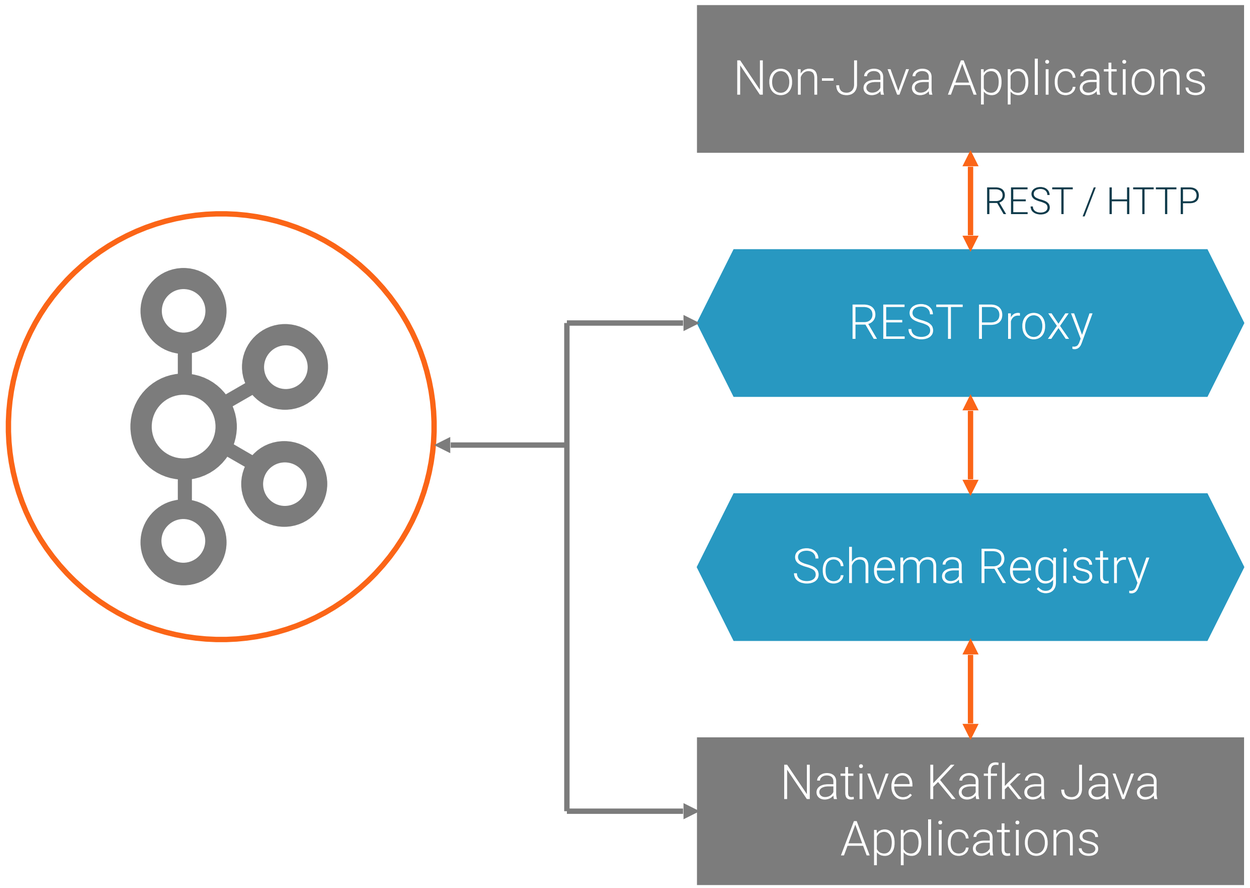
```
{
"connector.class": "io.confluent.connect.jdbc.JdbcSourceConnector",
"connection.url": "jdbc:mysql://localhost:3306/demo?user=rmoff&password=foo",
"table.whitelist": "sales,orders,customers"
}
```

https://docs.confluent.io/current/connect/

Kafka Connect



REST Proxy: Talking to Non-native Kafka Apps and Outside the Firewall



Provides a **RESTful interface** to a Kafka cluster

Simplifies message creation and consumption

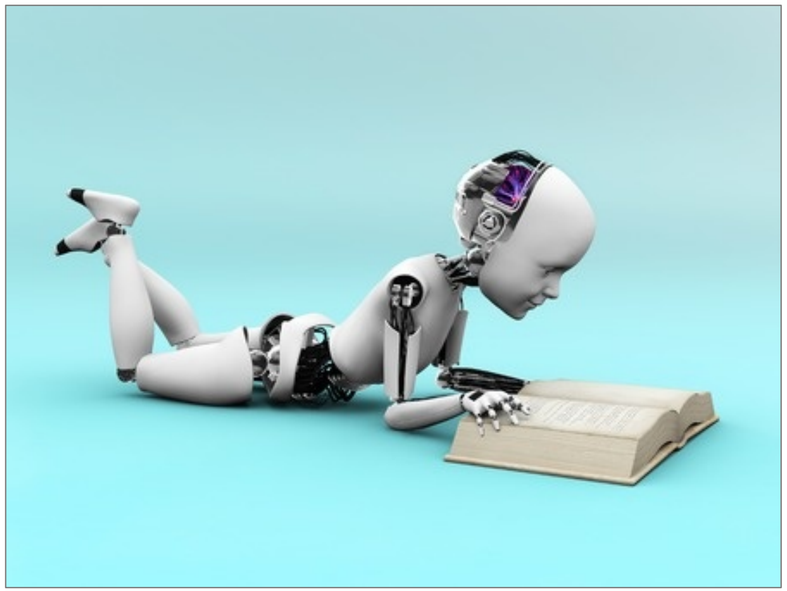
Simplifies administrative actions

What we will talk about…

- What is Apache Kafka?
- IoT and Kafka
- **Machine Learning and Kafka**
- Stream Processing with Kafka
- Put it all together :
  IoT + Stream Processing + ML + Kafka!

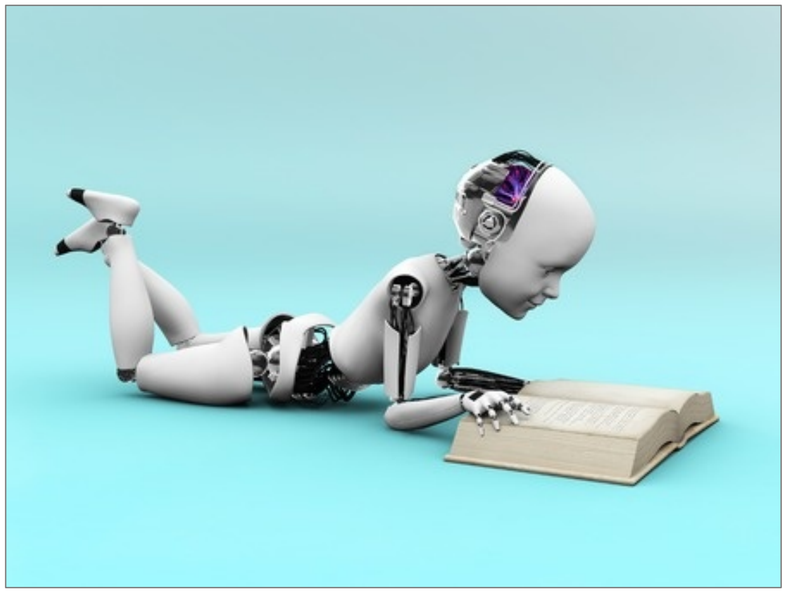
Machine Learning

... allows computers to find hidden insights without being explicitly programmed where to look.



Machine Learning

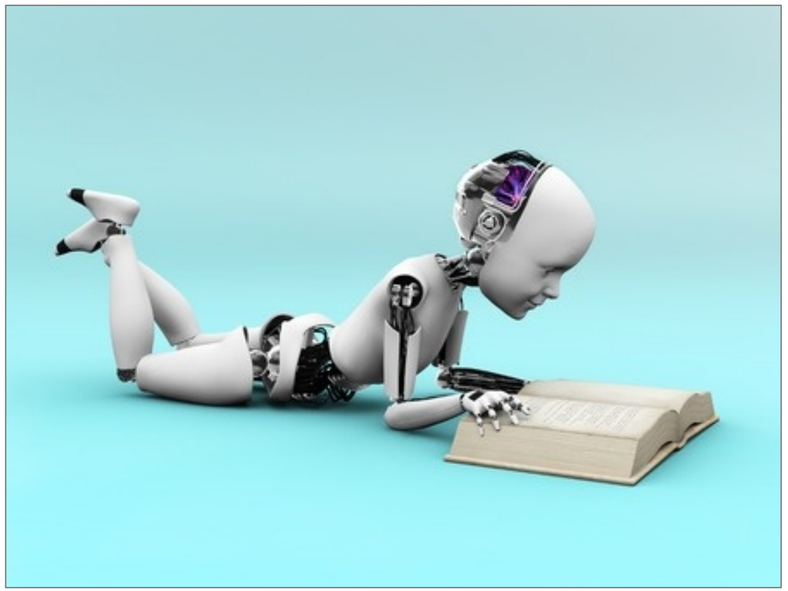
... allows computers to find hidden insights without being explicitly programmed where to look.

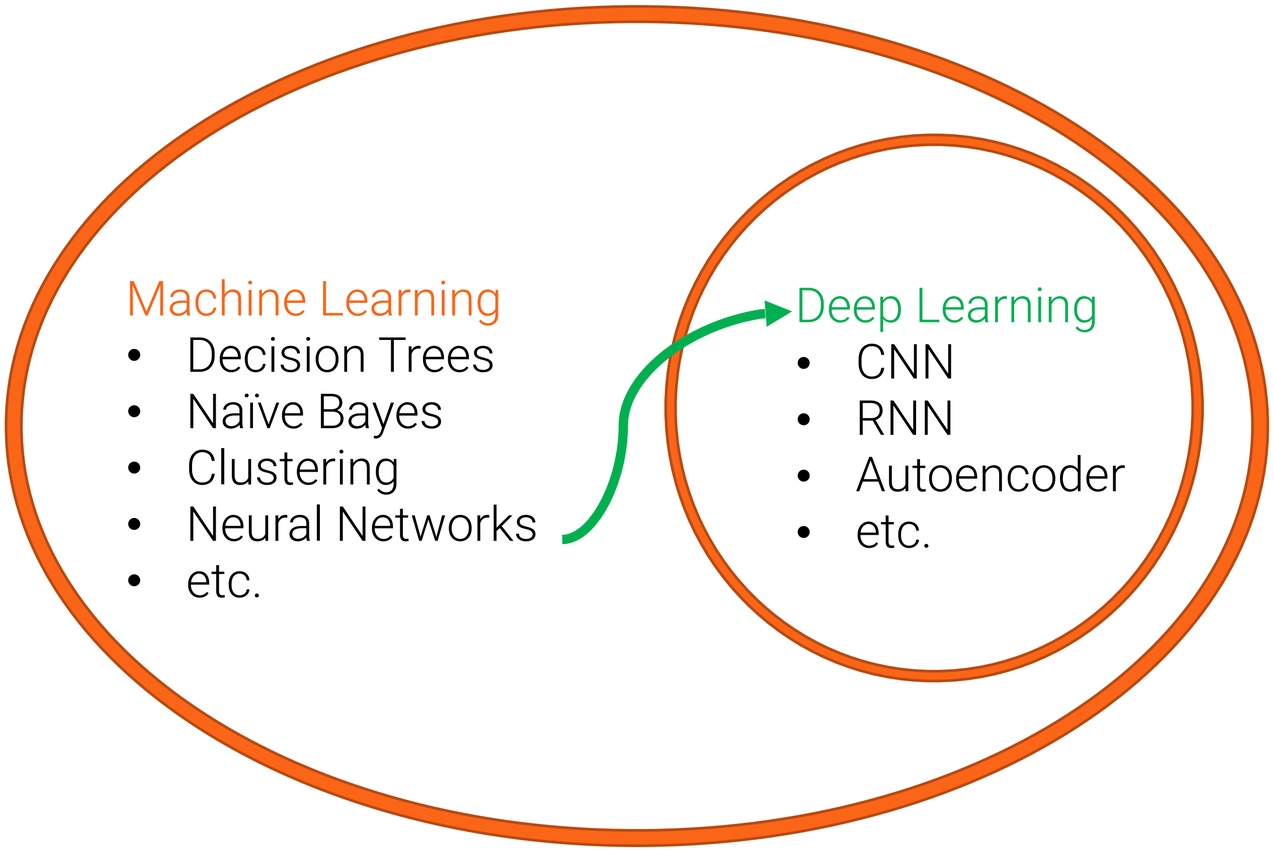


Machine Learning
- Decision Trees
- Naïve Bayes
- Clustering
- Neural Networks
- etc.

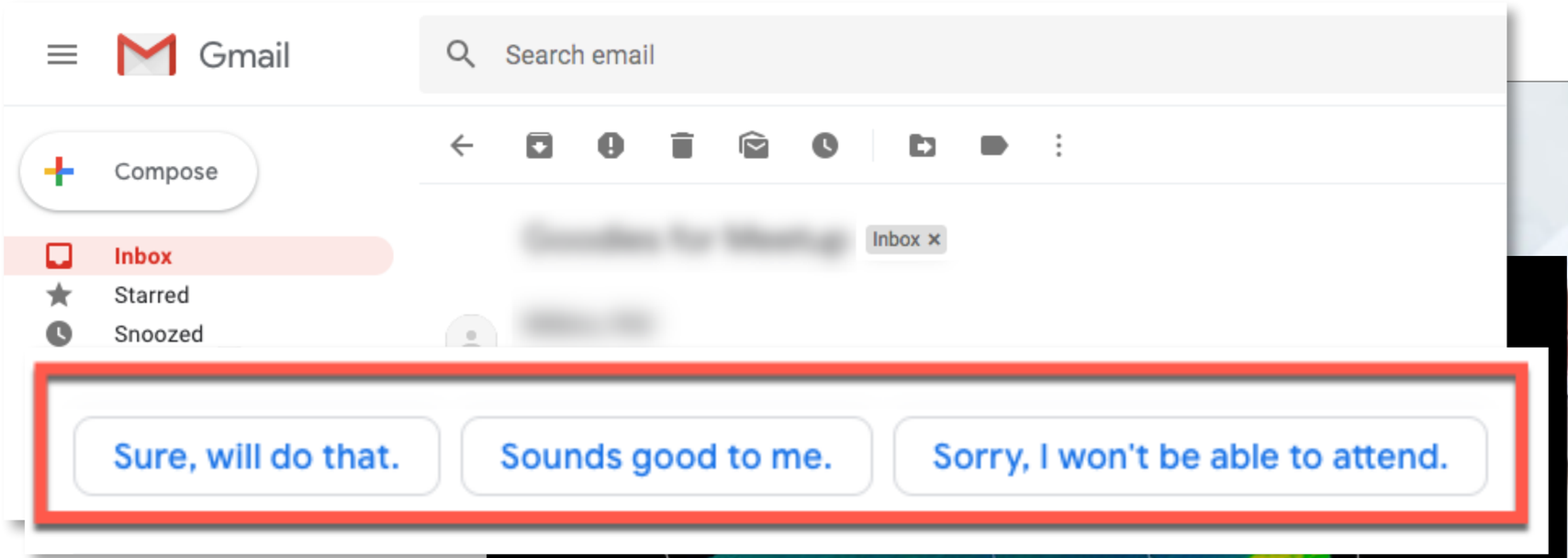
Machine Learning

... allows computers to find hidden insights without being explicitly programmed where to look.





Real World Examples of Machine Learning



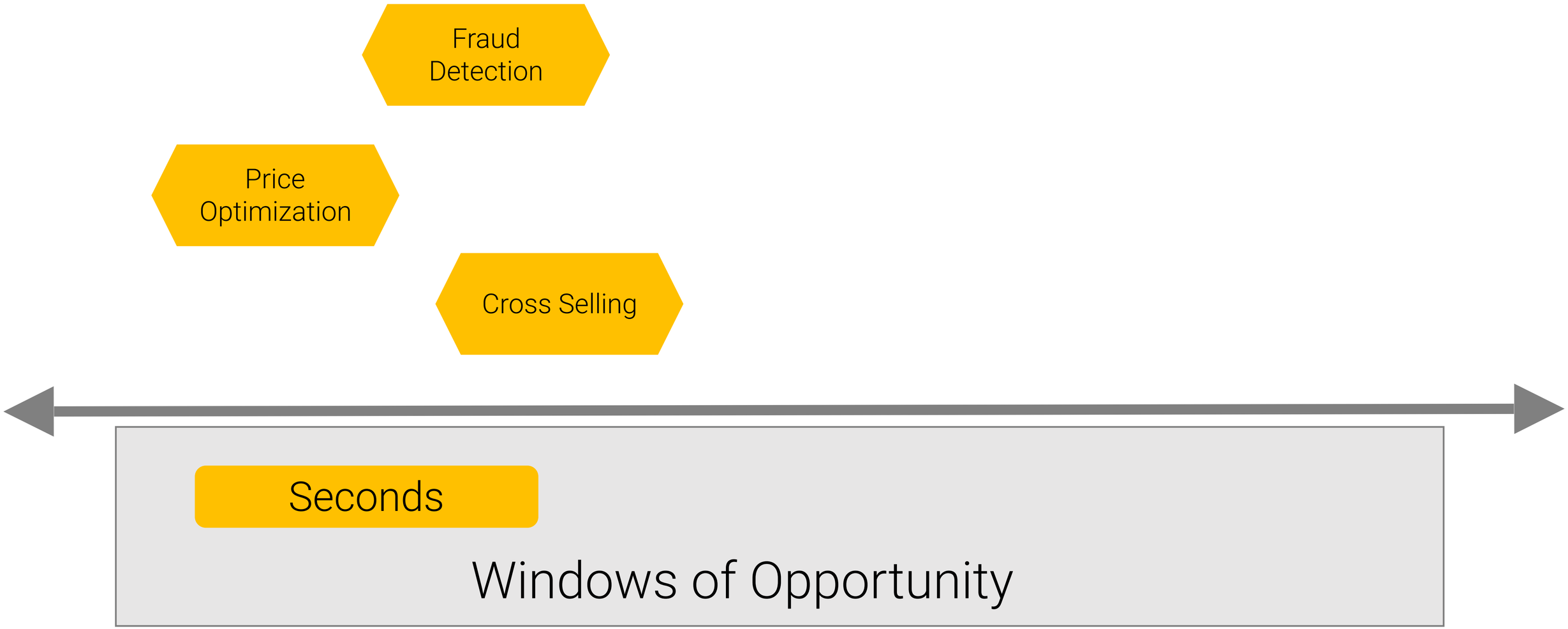
Automated dialogue

Healthcare screening

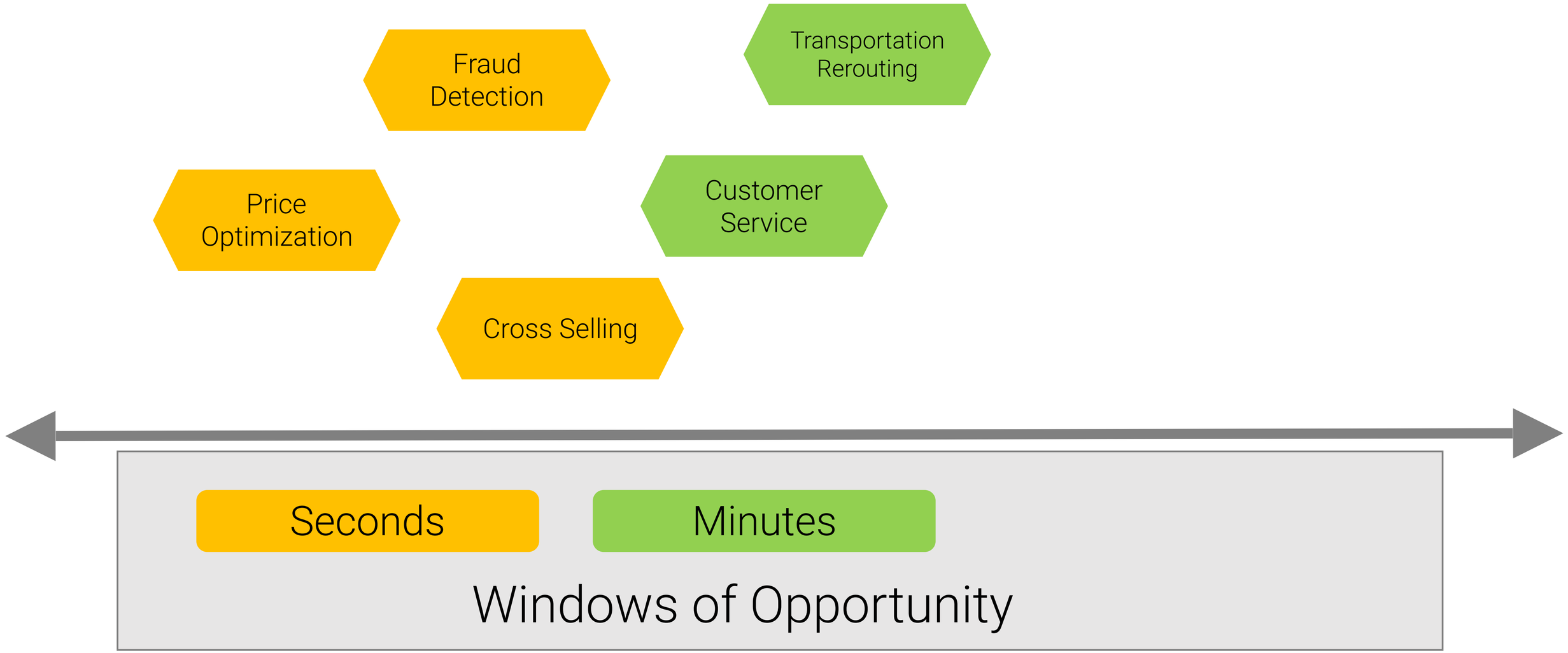
Search Results +
Product Recommendation

Leverage Machine Learning to Analyze and Act on Critical Business Moments

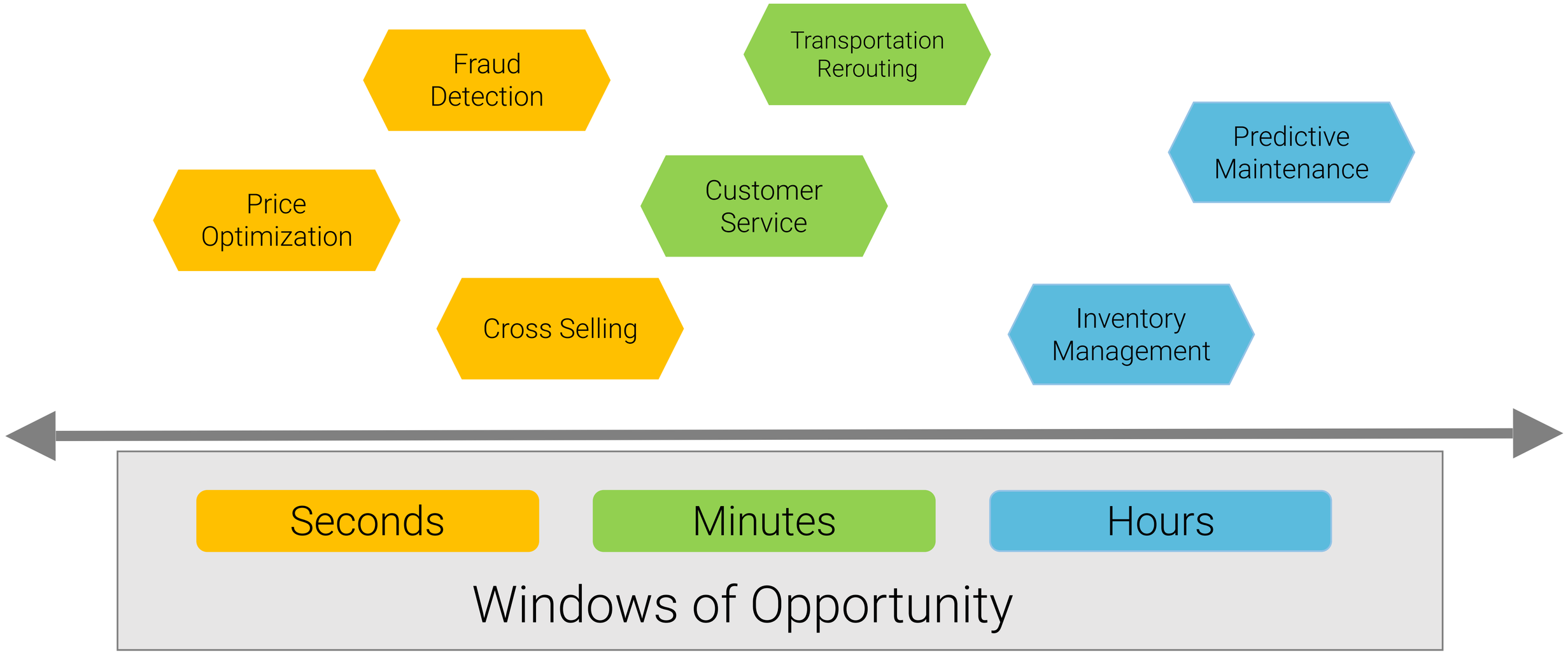
Leverage Machine Learning to Analyze and Act on Critical Business Moments



Leverage Machine Learning to Analyze and Act on Critical Business Moments



Leverage Machine Learning to Analyze and Act on Critical Business Moments



Languages, Frameworks, and Tools for Machine Learning

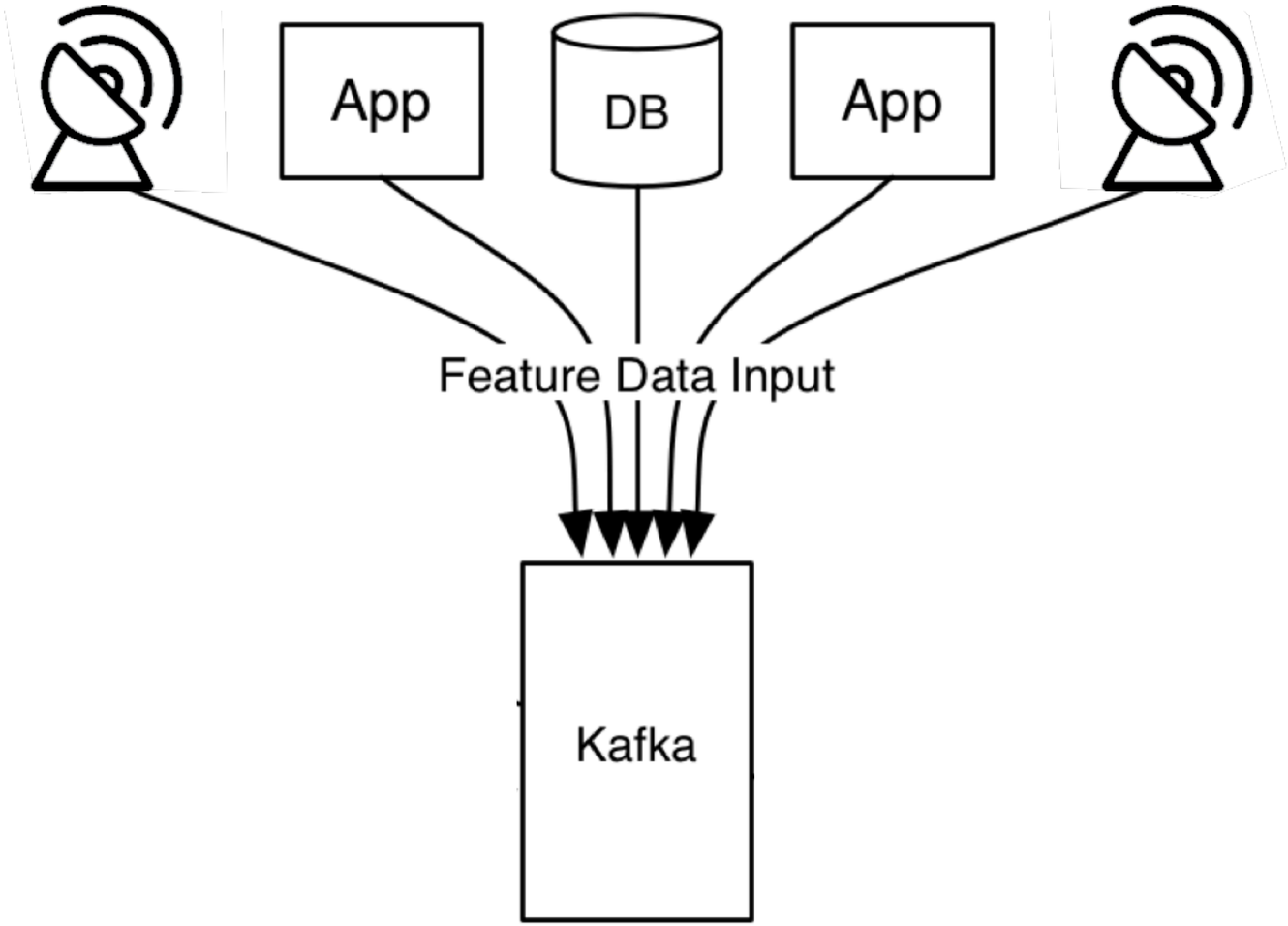
Portable Format
for Analytics (PFA)

Languages, Frameworks, and Tools for Machine Learning

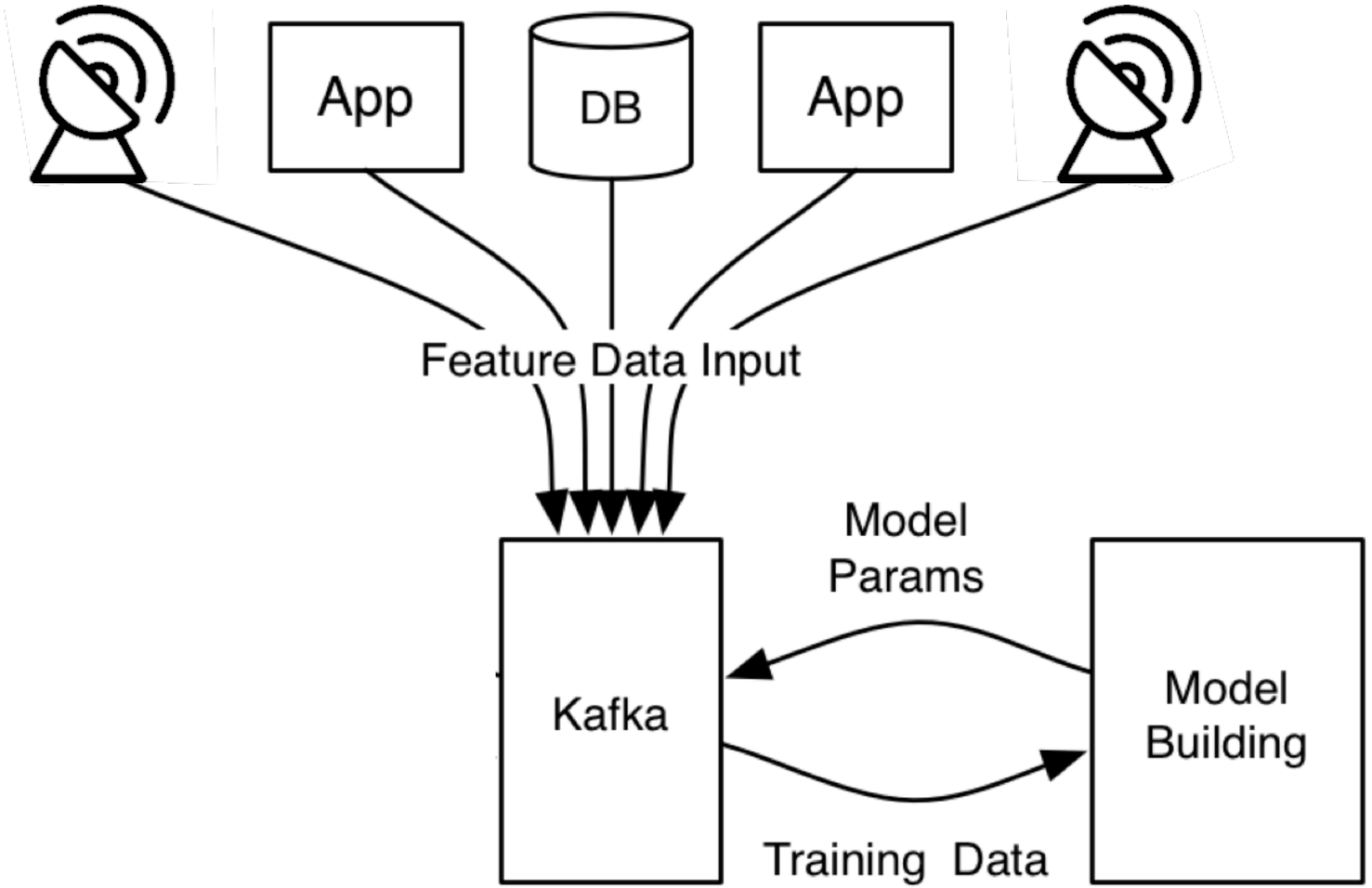
Portable Format
for Analytics (PFA)

There is no all-rounder → **ML-independent infrastructure needed!**

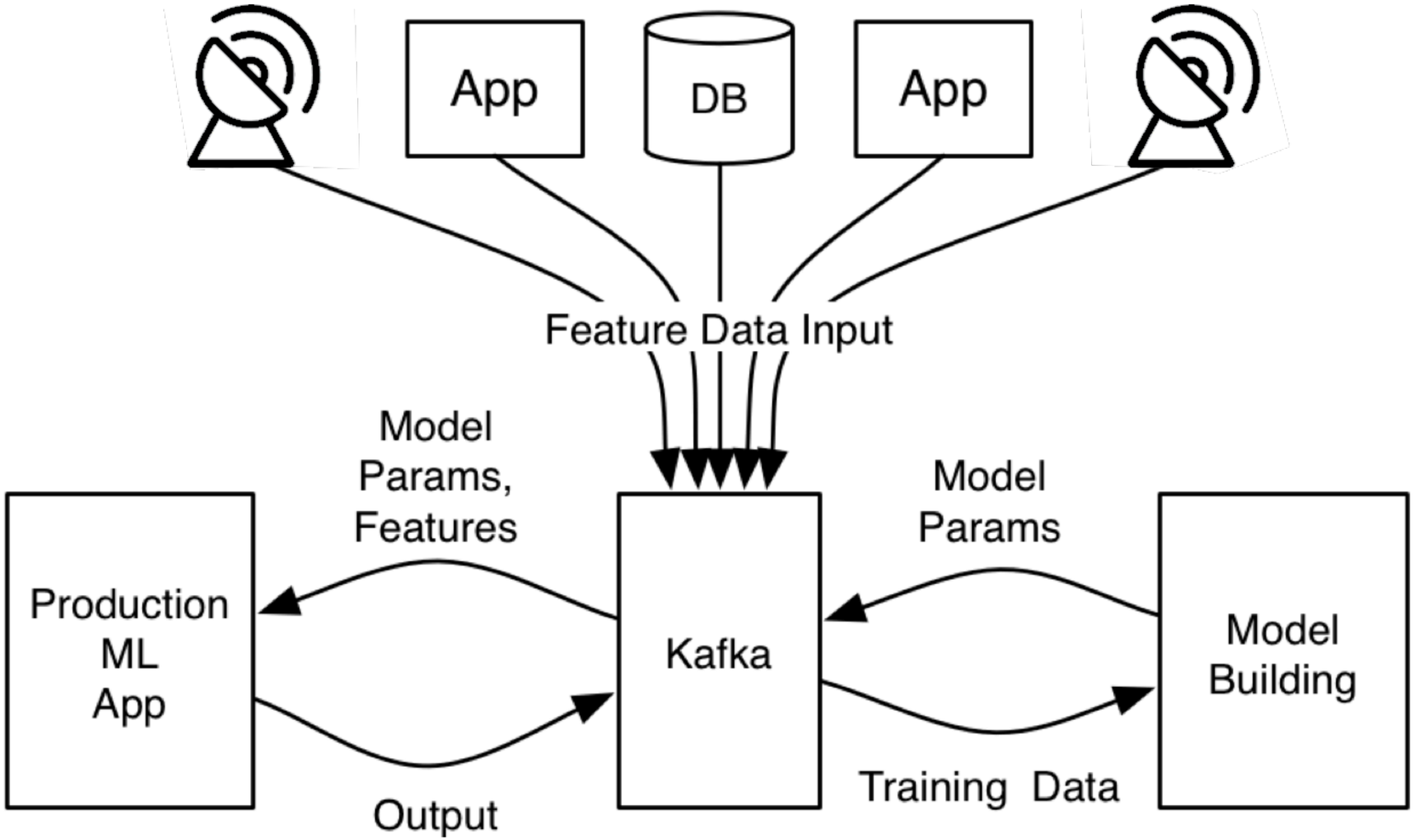
Apache Kafka's Open Source Ecosystem as Infrastructure for Machine Learning



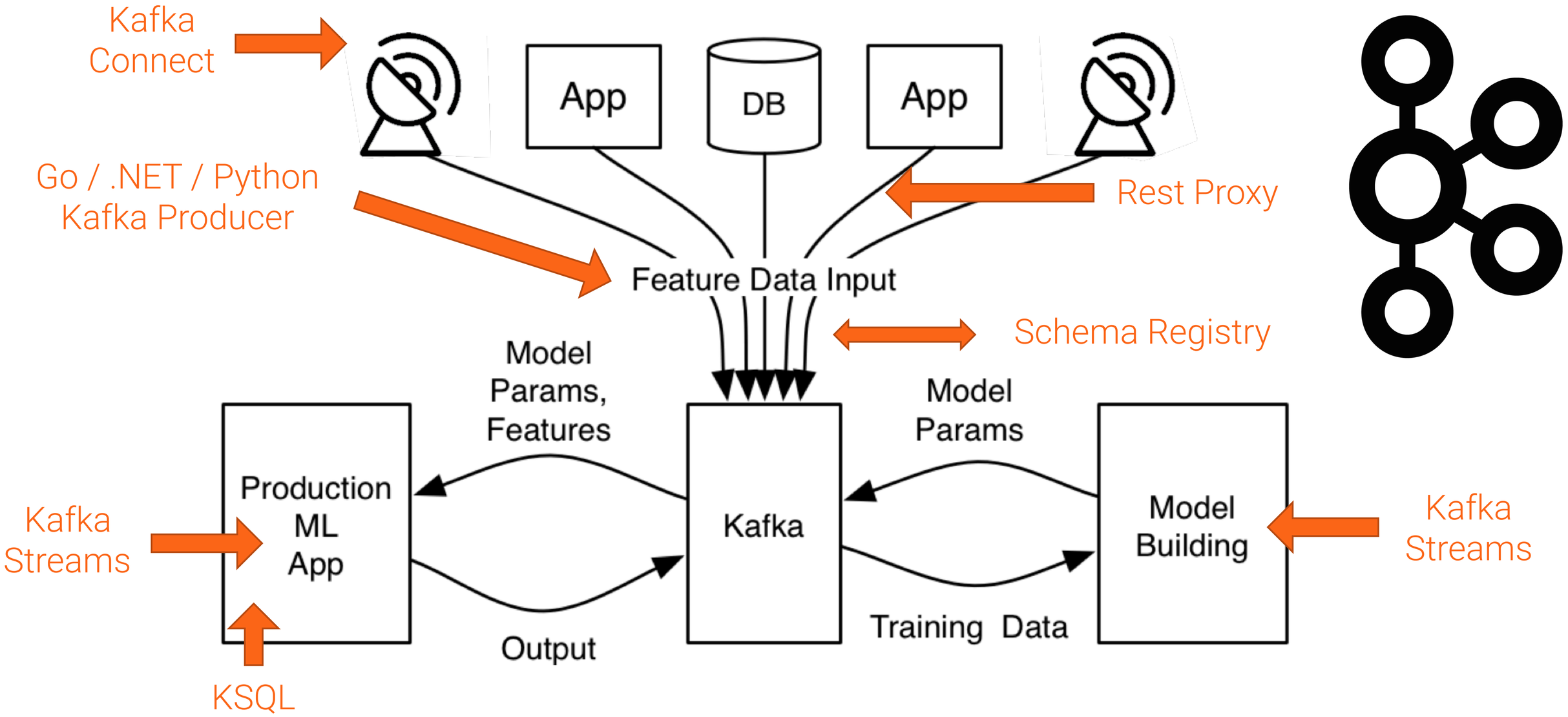
Apache Kafka's Open Source Ecosystem as Infrastructure for Machine Learning



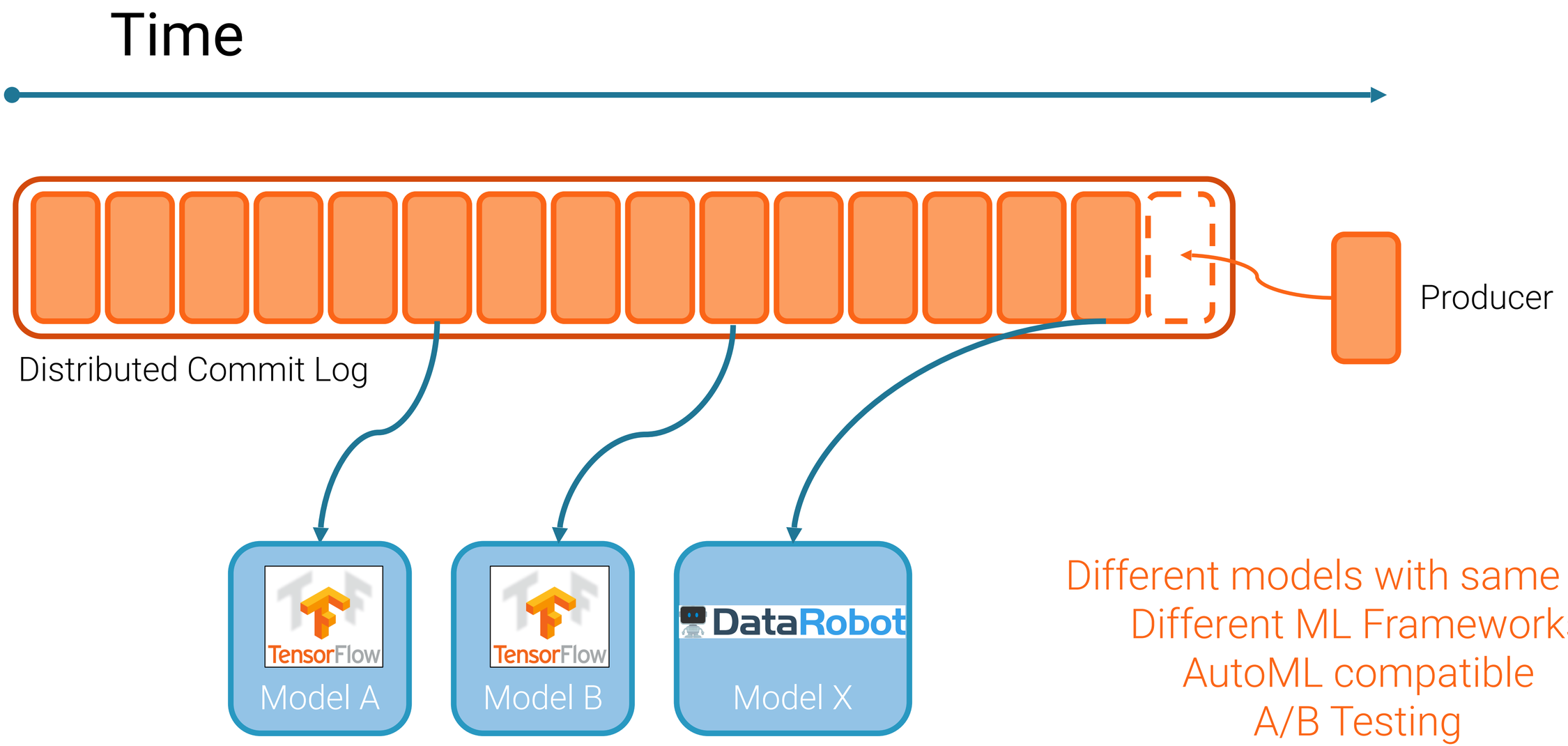
Apache Kafka's Open Source Ecosystem as Infrastructure for Machine Learning



Apache Kafka's Open Source Ecosystem as Infrastructure for Machine Learning



Replay-ability – A log never forgets!



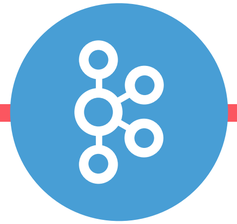
Different models with same data
Different ML Frameworks
AutoML compatible
A/B Testing

What we will talk about…

- What is Apache Kafka?
- IoT and Kafka
- Machine Learning and Kafka
- **Stream Processing with Kafka**
- Put it all together :

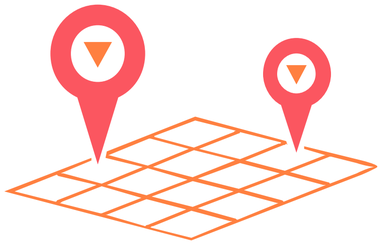
  IoT + Stream Processing + ML + Kafka!

the KAFKA STREAMS API is a JAVA API to BUILD REAL-TIME APPLICATIONS to POWER THE BUSINESS



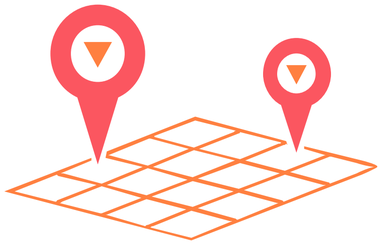
Things Kafka Streams Does

Things Kafka Streams Does

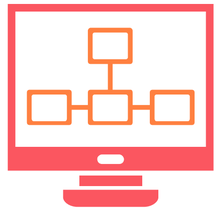


Runs everywhere

Things Kafka Streams Does

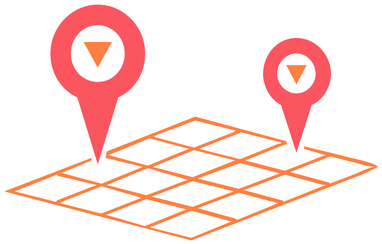


Runs everywhere

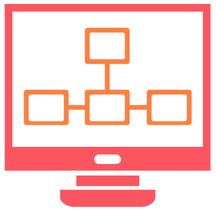


Clustering done for you

Things Kafka Streams Does



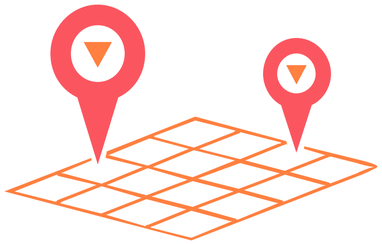
Runs everywhere



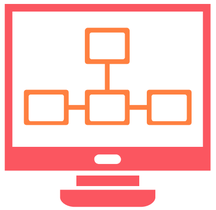
Clustering done for you

Exactly-once processing

Things Kafka Streams Does



Runs everywhere

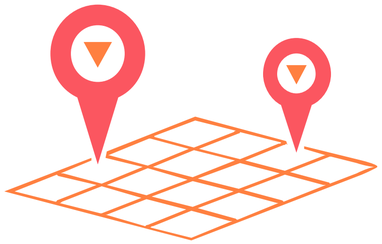


Clustering done for you

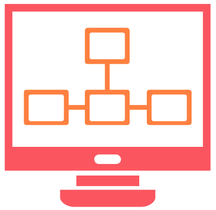
Exactly-once processing

Event-time processing

Things Kafka Streams Does



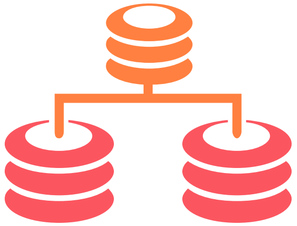
Runs everywhere



Clustering done for you

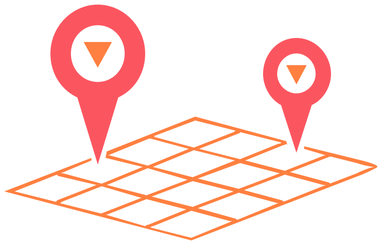
Exactly-once processing

Event-time processing

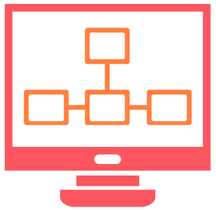


Integrated database

Things Kafka Streams Does



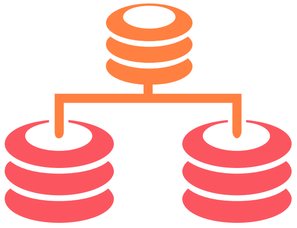
Runs everywhere



Clustering done for you

Exactly-once processing

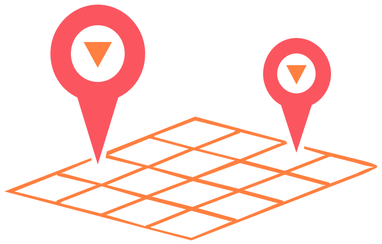
Event-time processing



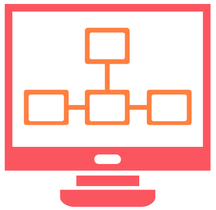
Integrated database

Joins, windowing, aggregation

Things Kafka Streams Does



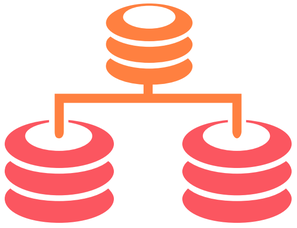
Runs everywhere



Clustering done for you

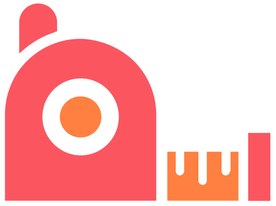
Exactly-once processing

Event-time processing

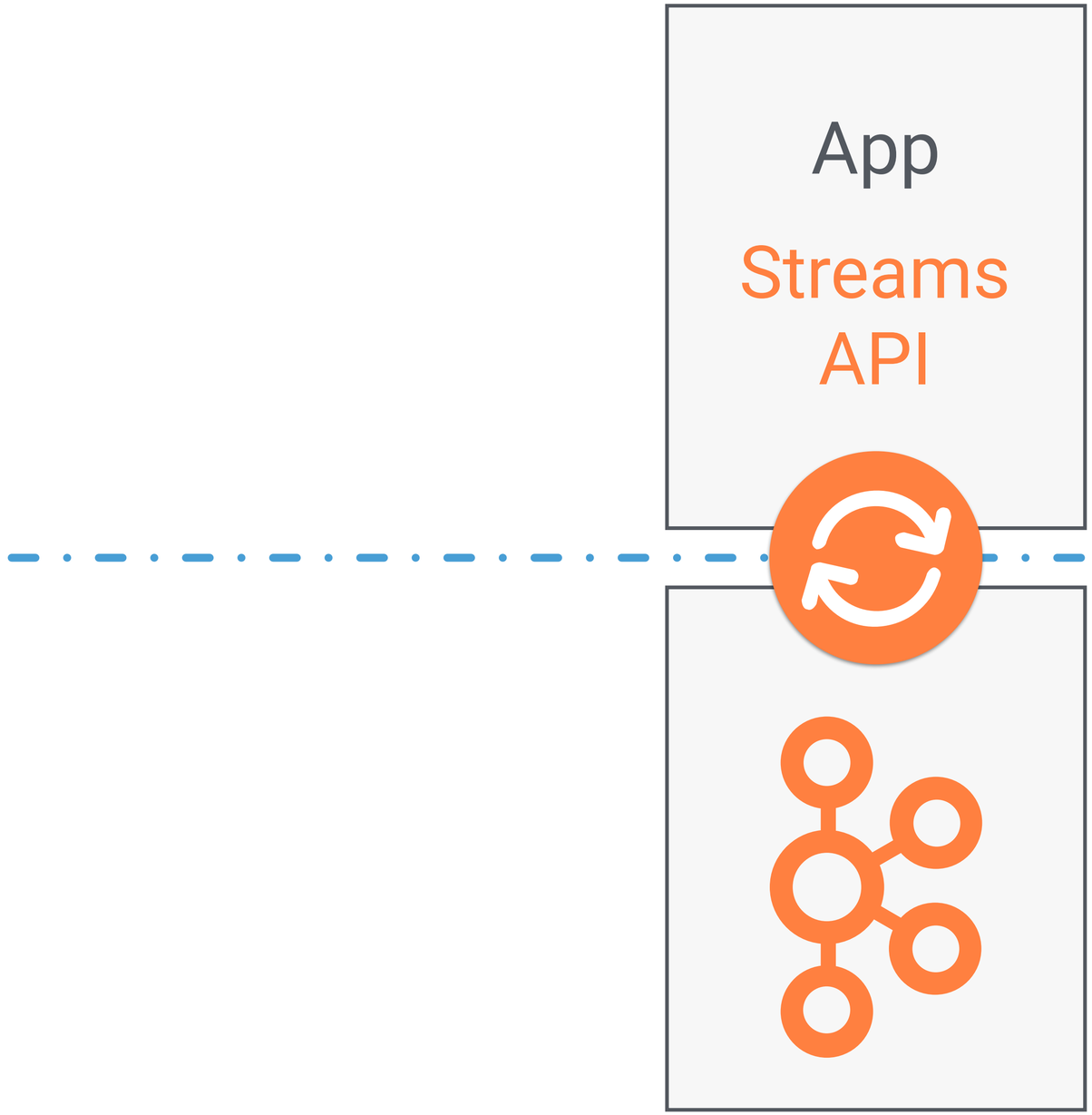


Integrated database

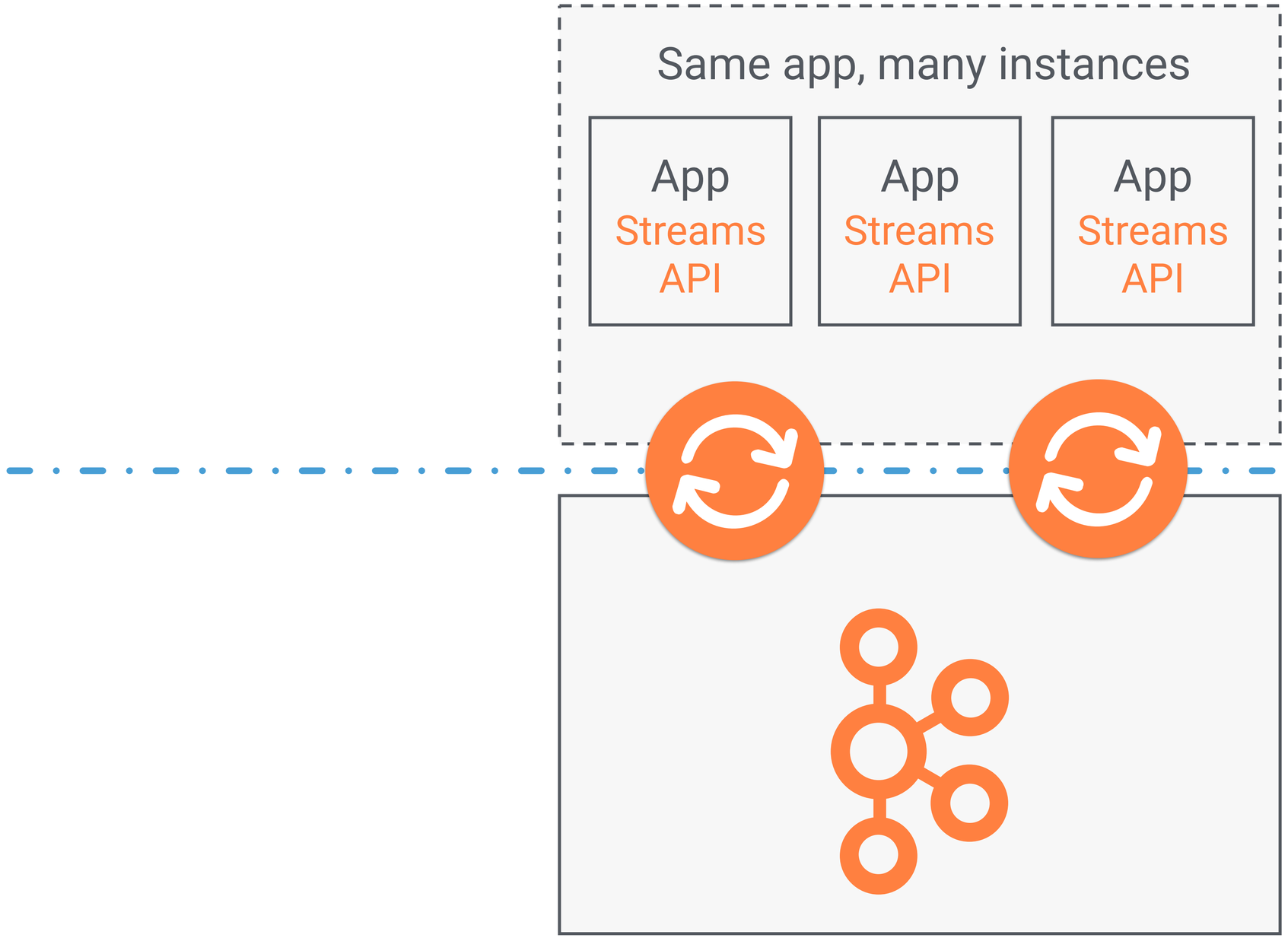
Joins, windowing, aggregation



S/M/L/XL/XXL/XXXL sizes



Not running inside brokers!

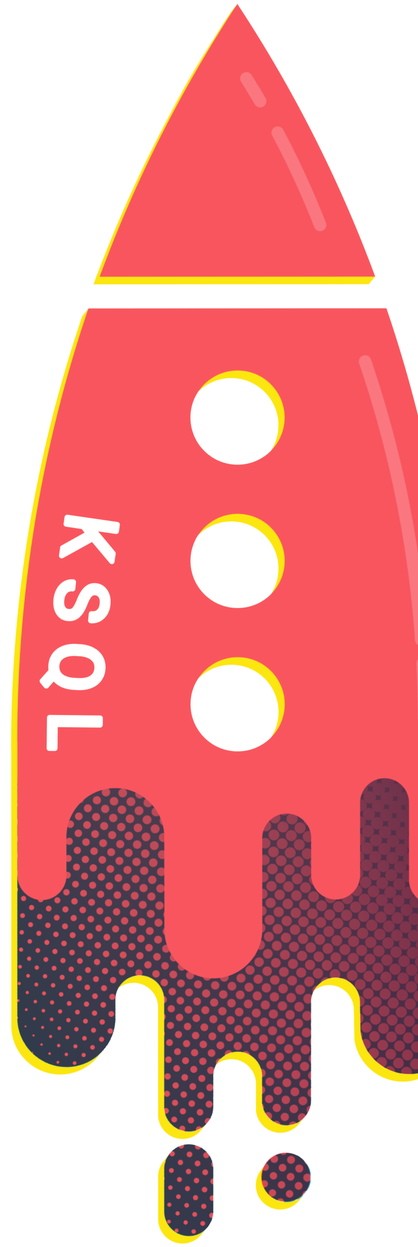


Brokers?
Nope!

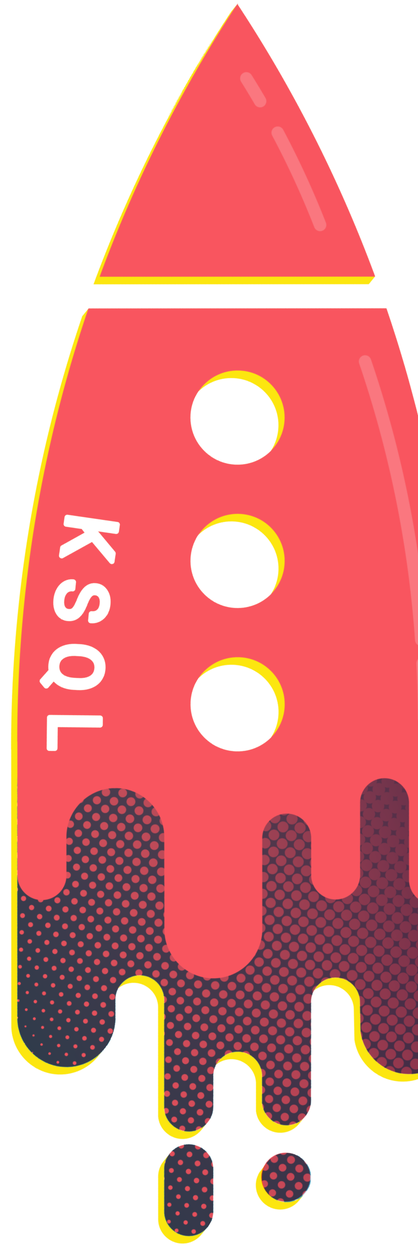
KSQL

is a

Declarative Stream Processing Language

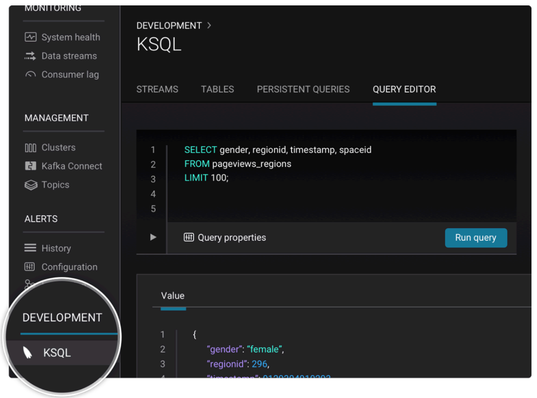


KSQL
is the
Streaming
SQL Engine
for
Apache Kafka



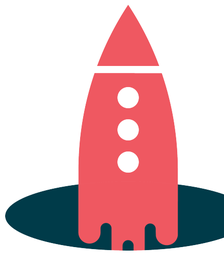
KSQL in Development and Production

Interactive KSQL
for development and testing



REST

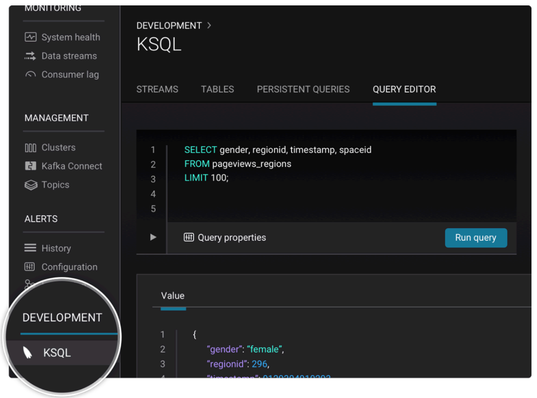
"Hmm, let me try out this idea..."



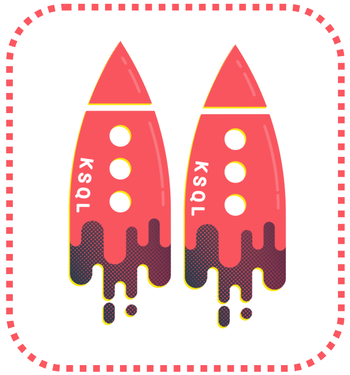
KSQL in Development and Production

Interactive KSQL
for development and testing

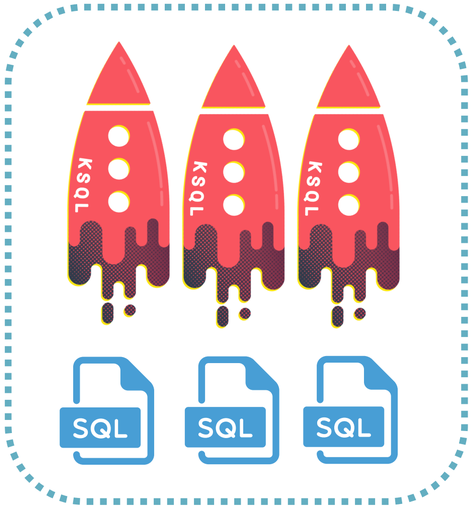
Headless KSQL
for Production



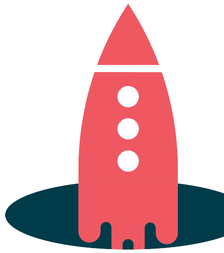
REST



Desired KSQL queries have been identified



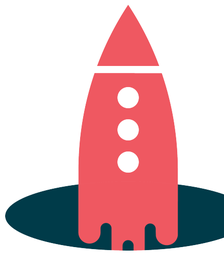
"Hmm, let me try out this idea..."



KSQL for Real-Time Monitoring

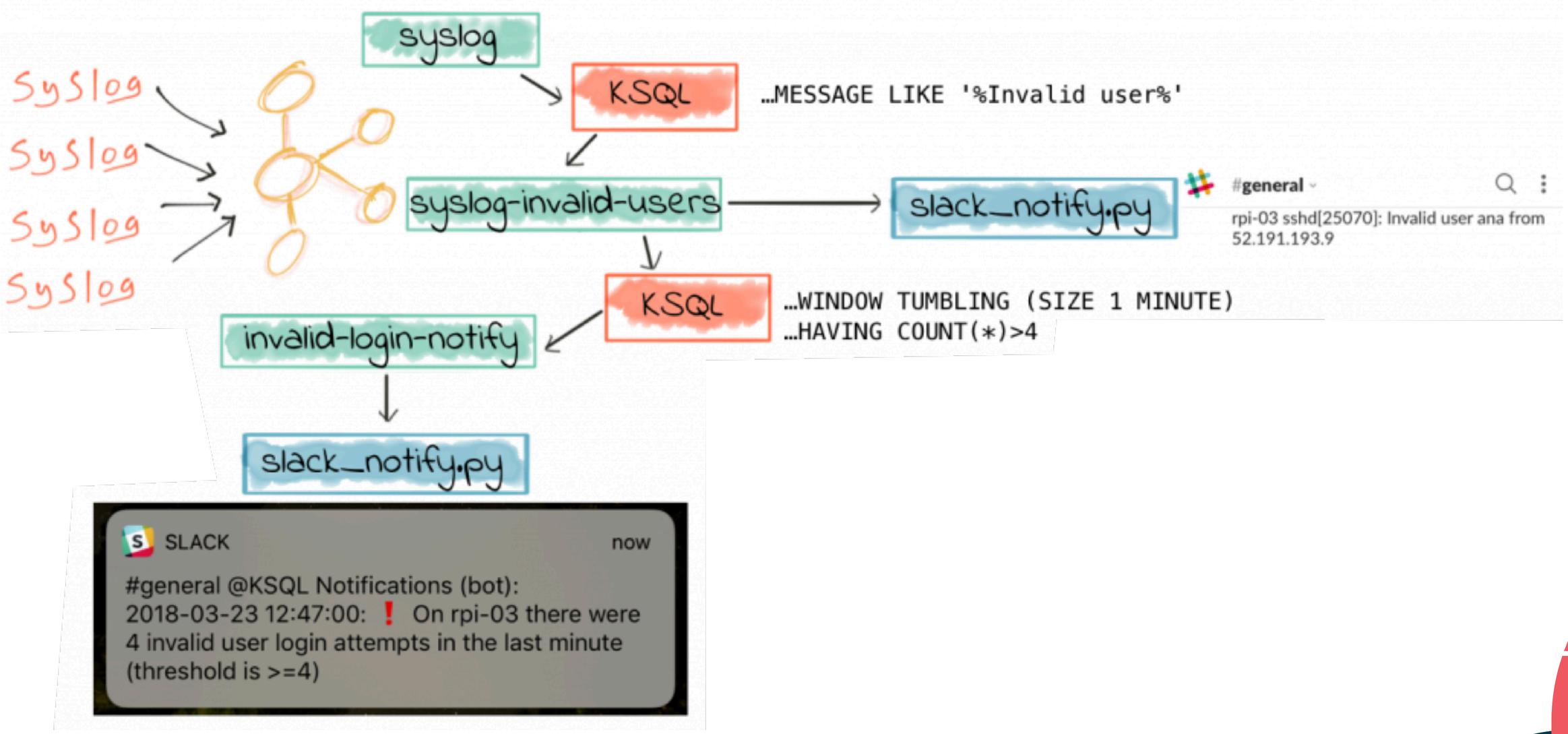
- Log data monitoring, tracking and alerting
- syslog data
- Sensor / IoT data

```
CREATE STREAM SYSLOG_INVALID_USERS AS
SELECT HOST, MESSAGE
FROM SYSLOG
WHERE MESSAGE LIKE '%Invalid user%';
```



http://cnfl.io/syslogs-filtering / http://cnfl.io/syslog-alerting

KSQL for Real-Time Monitoring

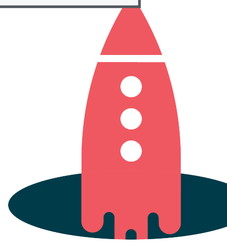


http://cnfl.io/syslog-alerting

KSQL for Streaming ETL

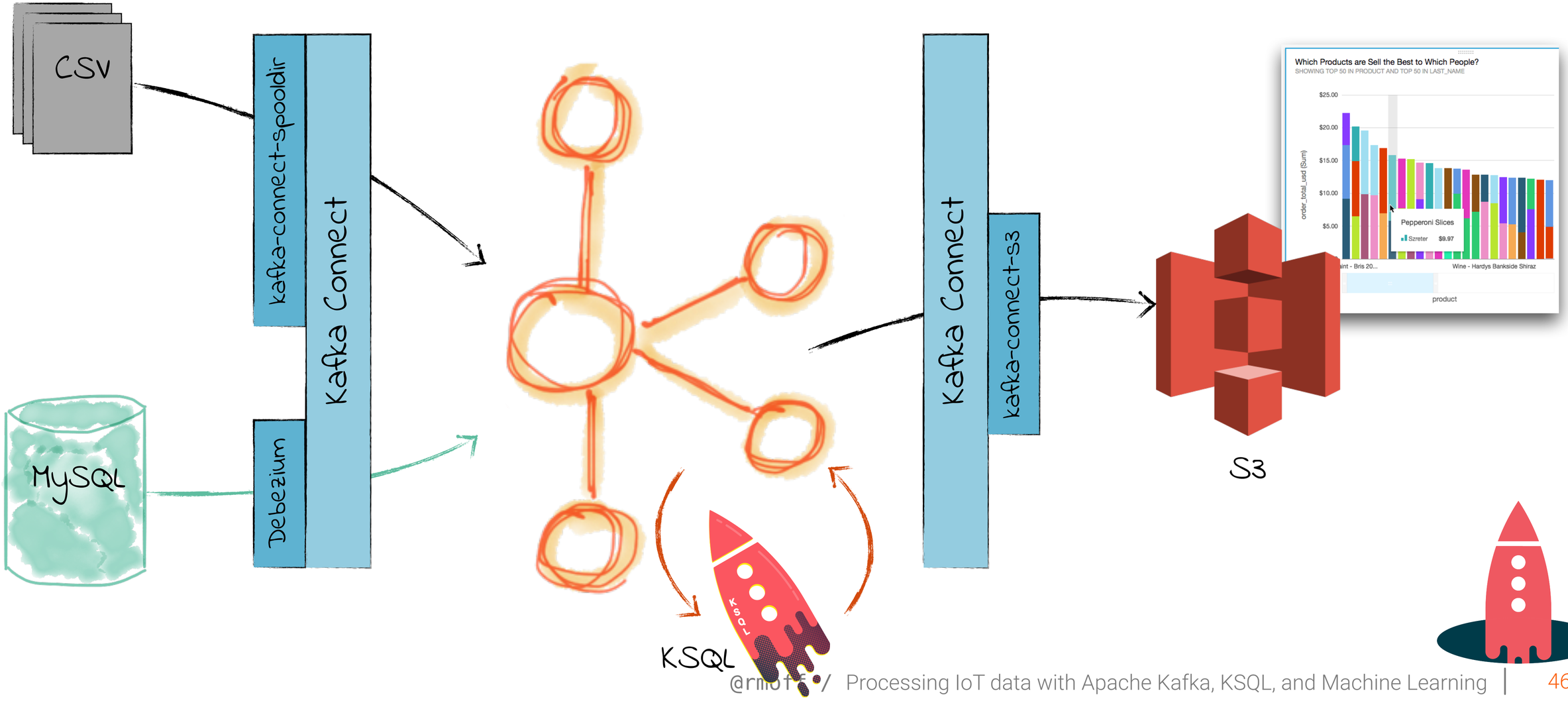
Joining, filtering, and aggregating streams of event data

```
CREATE STREAM vip_actions AS
  SELECT userid, page, action
  FROM clickstream c
  LEFT JOIN users u
    ON c.userid = u.user_id
  WHERE u.level = 'Platinum';
```



KSQL for Streaming ETL

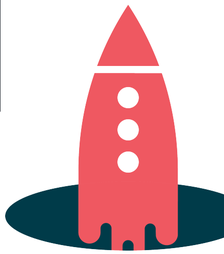
Denormalise streams of data from hetrogenous sources



KSQL for Anomaly Detection

Identifying patterns or anomalies in real-time data, surfaced in milliseconds

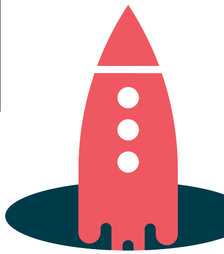
```
CREATE TABLE possible_fraud AS
  SELECT card_number, count(*)
    FROM authorization_attempts
    WINDOW TUMBLING (SIZE 5 SECONDS)
    GROUP BY card_number
    HAVING count(*) > 3;
```



KSQL for ML scoring

Apply Machine Learning models to live streams of data

```
CREATE STREAM possible_fraud AS
  SELECT txn_id, card_number
    FROM credit_card_txn_stream
   WHERE ANOMALY(txn_data) > 0.8;
```



What we will talk about…

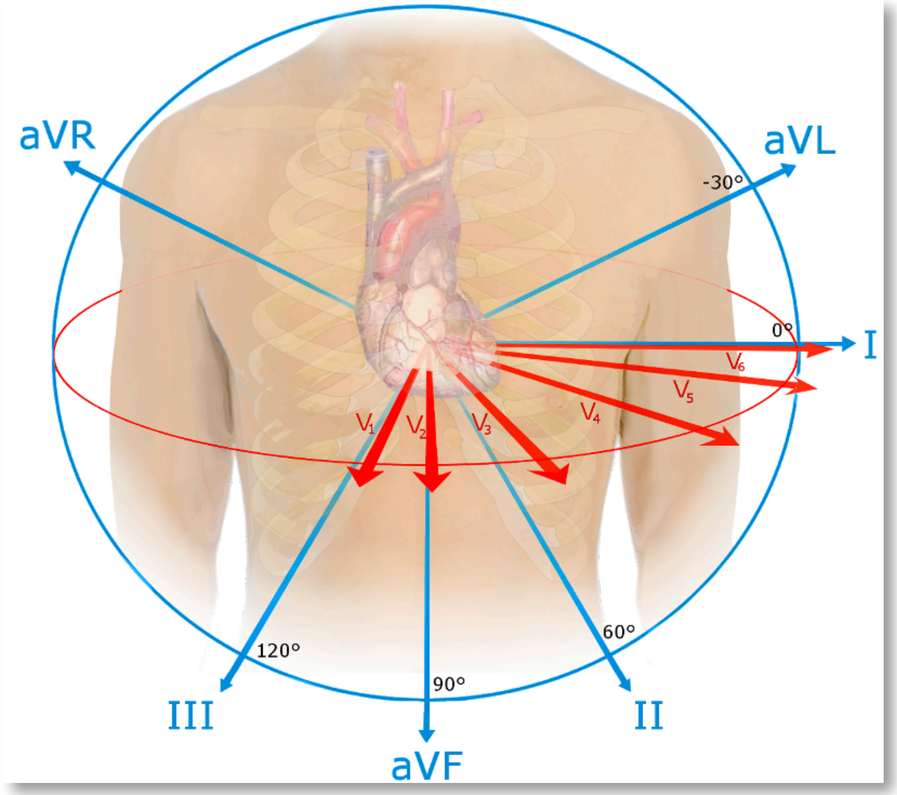
- What is Apache Kafka?
- IoT and Kafka
- Machine Learning and Kafka
- Stream Processing with Kafka
- **Put it all together :**
  IoT + Stream Processing + ML + Kafka!

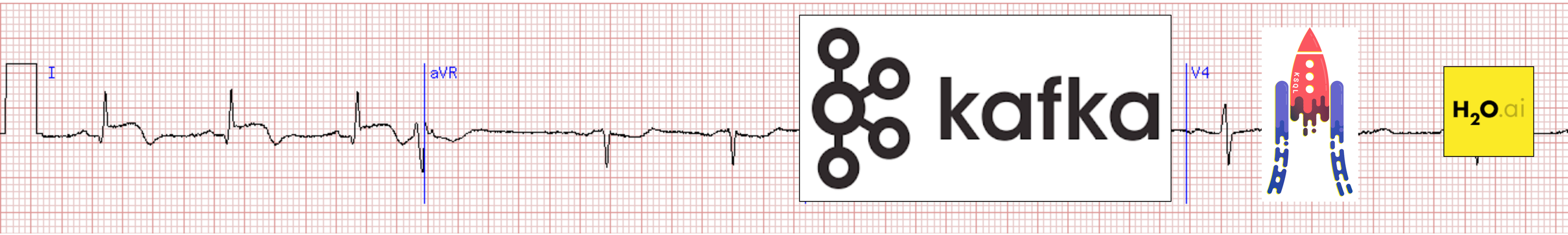
Live Demo – Prebuilt Model Embedded in KSQL Function

Use Case:
Anomaly Detection
(Sensor Healthcheck)

Machine Learning Algorithm:
Autoencoder, built with H2O

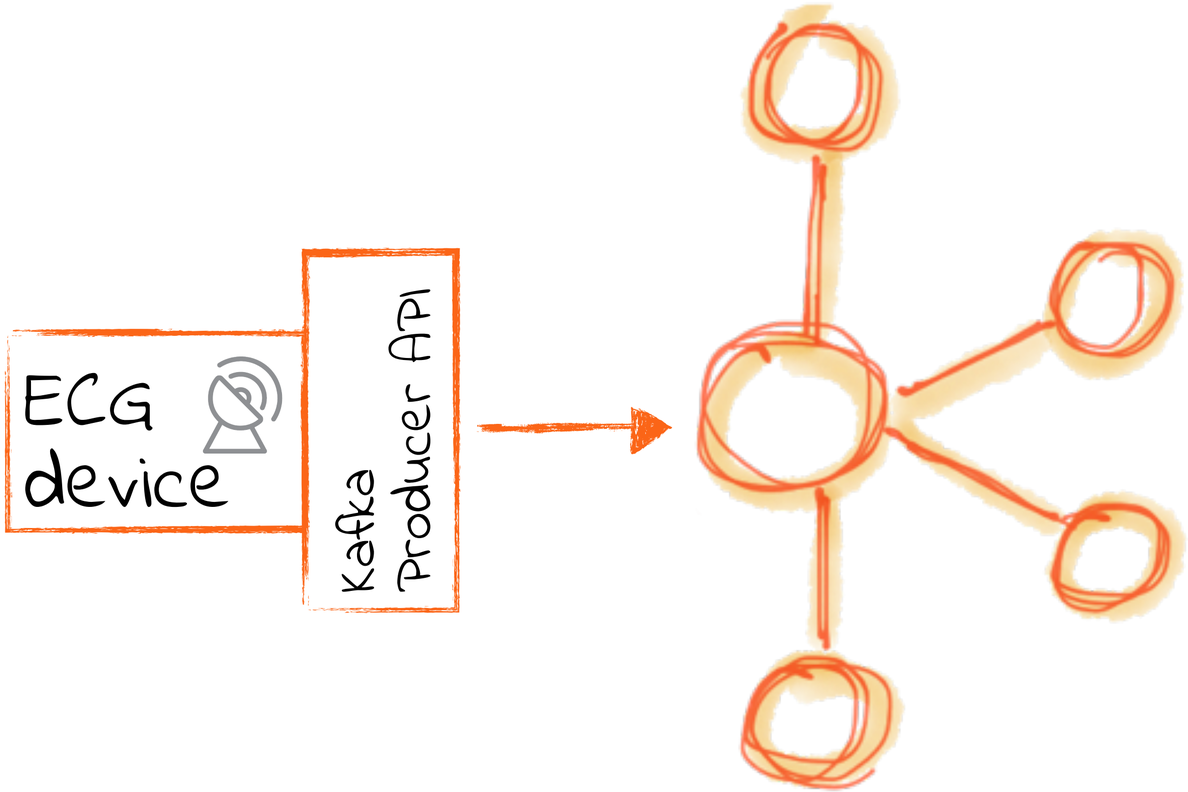
Streaming Platform:
Apache Kafka and KSQL



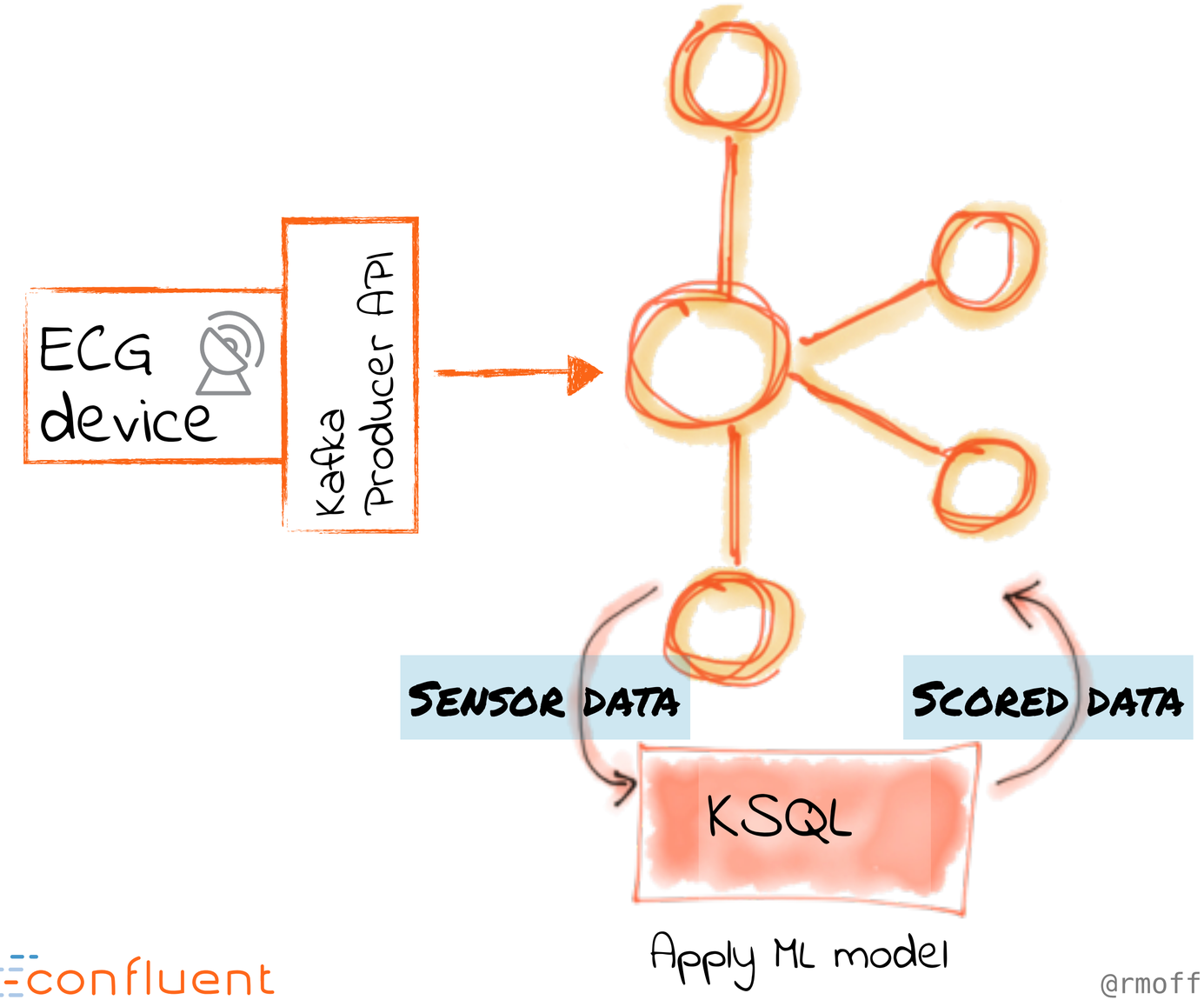


Overview

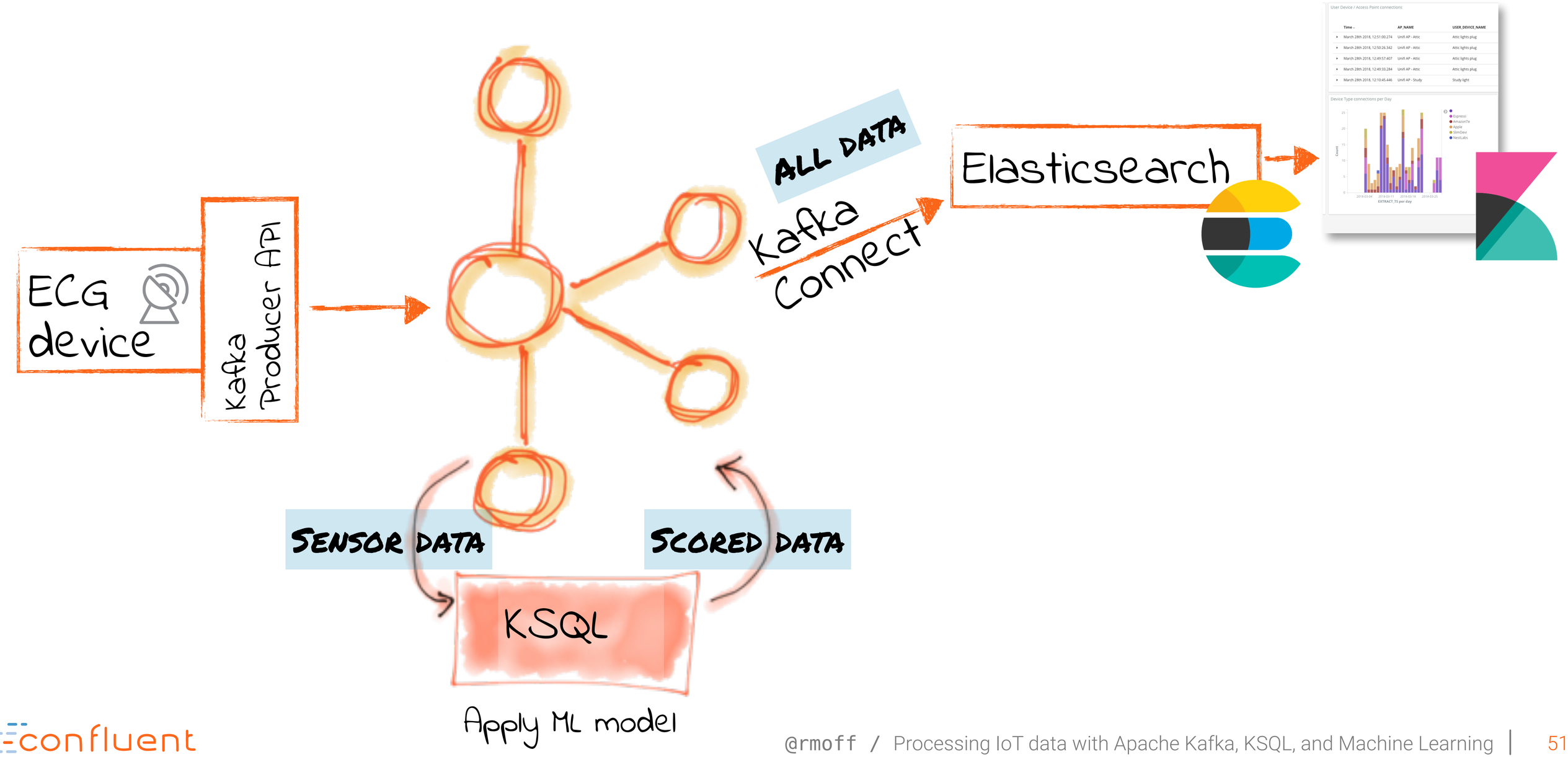
Overview



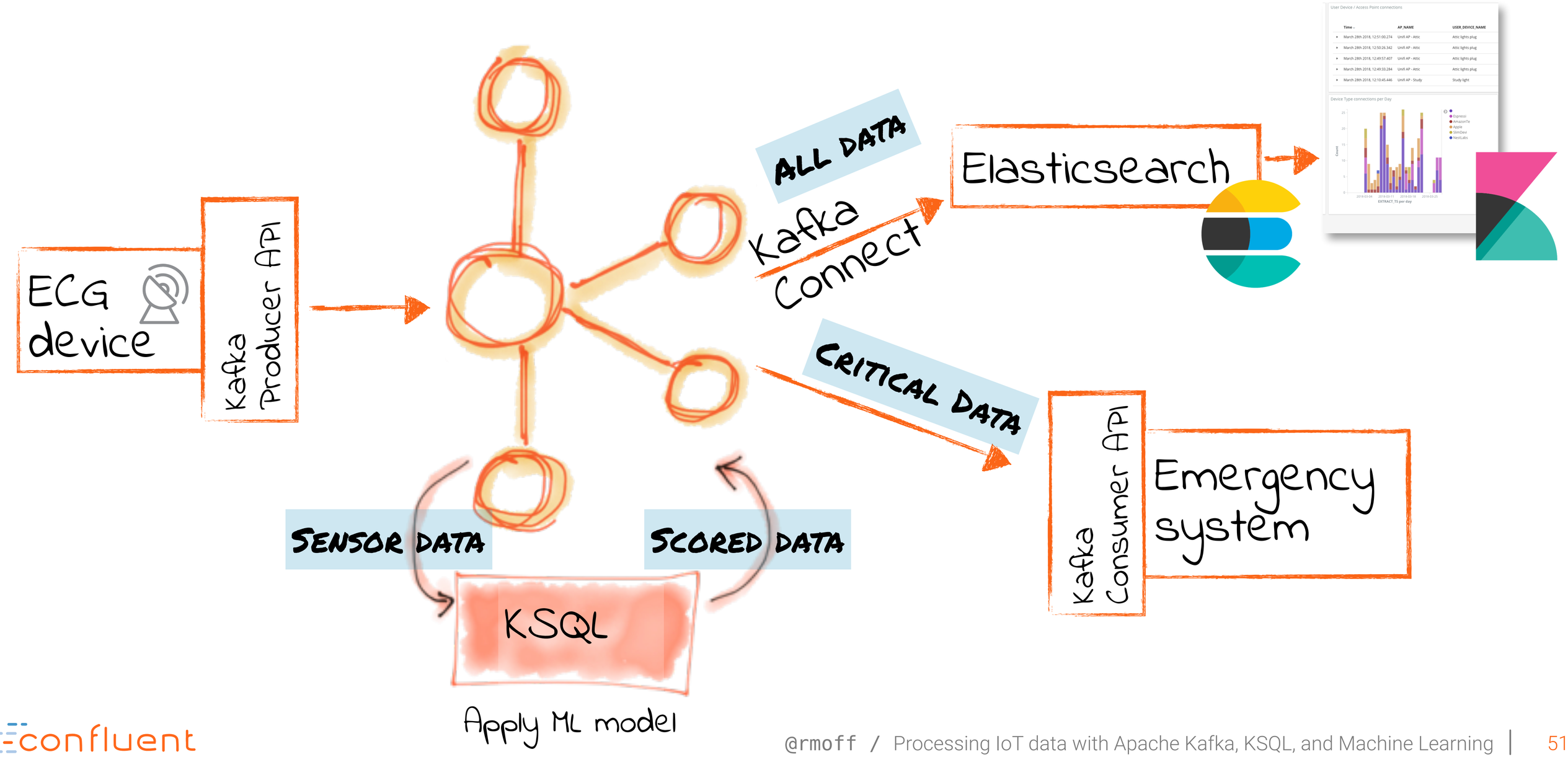
Overview


ECG device
Kafka Producer API
SENSOR DATA
SCORED DATA
KSQL
Apply ML model

Overview


ECG device
Kafka Producer API
ALL DATA
Kafka Connect
Elasticsearch
SENSOR DATA
SCORED DATA
KSQL
Apply ML model

Overview


ECG device
Kafka Producer API
ALL DATA
Kafka Connect
Elasticsearch
CRITICAL DATA
Kafka Consumer API
Emergency system
SENSOR DATA
SCORED DATA
KSQL
Apply ML model

Training the model

Training the model

```
anomaly_model <- h2o.deeplearning(
x = names(train_ecg),
training_frame =  train_ecg,
activation = "Tanh",
autoencoder = TRUE,
hidden = c(50,20,50),
sparse = TRUE,
l1 = 1e-4,
epochs = 100)
```

R

Training the model

```
anomaly_model <- h2o.deeplearning(
x = names(train_ecg),
training_frame =  train_ecg,
activation = "Tanh",
autoencoder = TRUE,
hidden = c(50,20,50),
sparse = TRUE,
l1 = 1e-4,
epochs = 100)
```

R

```
2.10,2.13,2.19,2.28,2.44,2.62,2.80,3.04,3.36,3.69,3.97,4.24,4.53,4.80,5.02,5.21,5.40,5.57,5.7
79,5.86,5.92,5.98,6.02,6.06,6.08,6.14,6.18,6.22,6.27,6.32,6.35,6.38,6.45,6.49,6.53,6.57,6.64,
,6.73,6.78,6.83,6.88,6.92,6.94,6.98,7.01,7.03,7.05,7.06,7.07,7.08,7.06,7.04,7.03,6.99,6.94,6.
.83,6.77,6.69,6.60,6.53,6.45,6.36,6.27,6.19,6.11,6.03,5.94,5.88,5.81,5.75,5.68,5.62,5.61,5.54
9,5.45,5.42,5.38,5.34,5.31,5.30,5.29,5.26,5.23,5.23,5.22,5.20,5.19,5.18,5.19,5.17,5.15,5.14,5
5.16,5.15,5.15,5.15,5.14,5.14,5.14,5.15,5.14,5.14,5.13,5.15,5.15,5.15,5.14,5.16,5.15,5.15,5.1
14,5.15,5.15,5.14,5.13,5.14,5.14,5.11,5.12,5.12,5.12,5.09,5.09,5.09,5.10,5.08,5.08,5.08,5.08,
,5.05,5.06,5.07,5.05,5.03,5.03,5.04,5.03,5.01,5.01,5.02,5.01,5.01,5.00,5.00,5.02,5.01,4.98,5.
.00,5.00,4.99,5.00,5.01,5.02,5.01,5.03,5.03,5.02,5.02,5.04,5.04,5.04,5.02,5.02,5.01,4.99,4.98
6,4.96,4.96,4.94,4.93,4.93,4.93,4.93,4.93,5.02,5.27,5.80,5.94,5.58,5.39,5.32,5.25,5.21,5.13,4
4.71,4.39,4.05,3.69,3.32,3.05,2.99,2.74,2.61,2.47,2.35,2.26,2.20,2.15,2.10,2.08
```

Training data

Training the model

```
anomaly_model <- h2o.deeplearning(
x = names(train_ecg),
training_frame =  train_ecg,
activation = "Tanh",
autoencoder = TRUE,
hidden = c(50,20,50),
sparse = TRUE,
l1 = 1e-4,
epochs = 100)
```

R

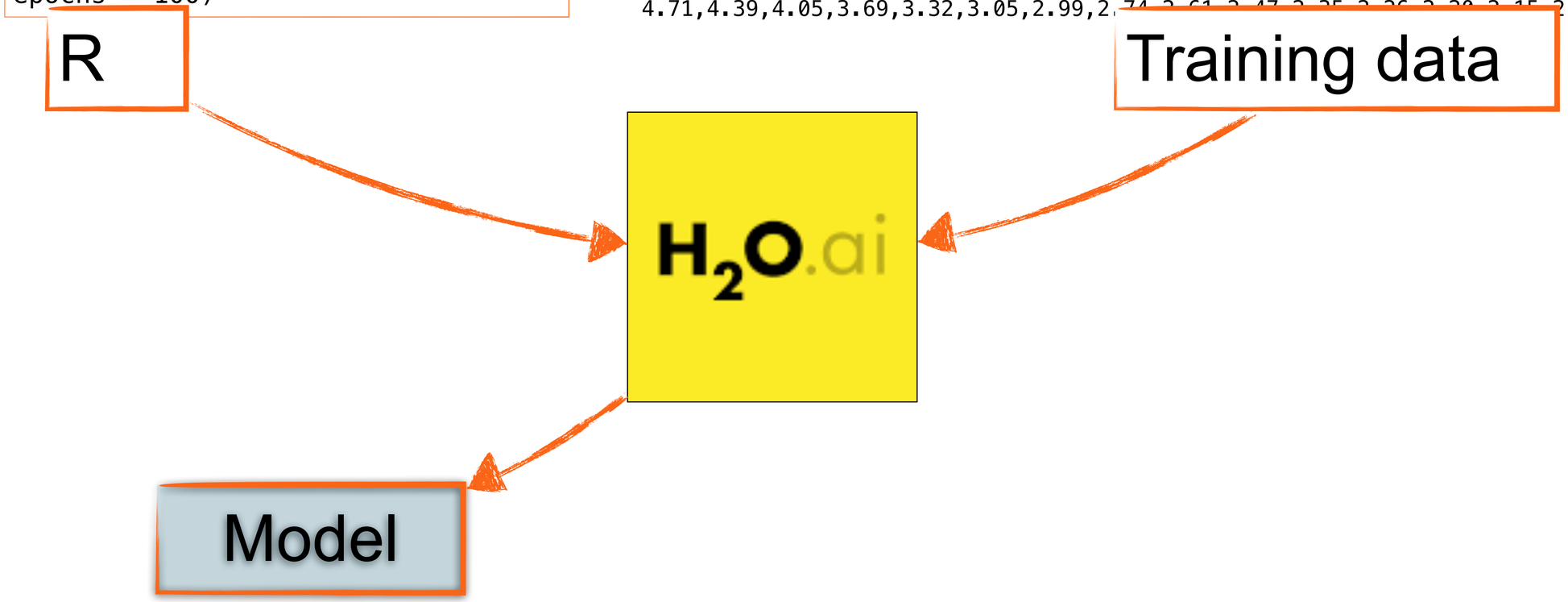
```
2.10,2.13,2.19,2.28,2.44,2.62,2.80,3.04,3.36,3.69,3.97,4.24,4.53,4.80,5.02,5.21,5.40,5.57,5.7
79,5.86,5.92,5.98,6.02,6.06,6.08,6.14,6.18,6.22,6.27,6.32,6.35,6.38,6.45,6.49,6.53,6.57,6.64,
,6.73,6.78,6.83,6.88,6.92,6.94,6.98,7.01,7.03,7.05,7.06,7.07,7.08,7.06,7.04,7.03,6.99,6.94,6.
.83,6.77,6.69,6.60,6.53,6.45,6.36,6.27,6.19,6.11,6.03,5.94,5.88,5.81,5.75,5.68,5.62,5.61,5.54
9,5.45,5.42,5.38,5.34,5.31,5.30,5.29,5.26,5.23,5.23,5.22,5.20,5.19,5.18,5.19,5.17,5.15,5.14,5
5.16,5.15,5.15,5.15,5.14,5.14,5.14,5.15,5.14,5.14,5.13,5.15,5.15,5.15,5.14,5.16,5.15,5.15,5.1
14,5.15,5.15,5.14,5.13,5.14,5.14,5.11,5.12,5.12,5.12,5.09,5.09,5.09,5.10,5.08,5.08,5.08,5.08,
,5.05,5.06,5.07,5.05,5.03,5.03,5.04,5.03,5.01,5.01,5.02,5.01,5.01,5.00,5.00,5.02,5.01,4.98,5.
.00,5.00,4.99,5.00,5.01,5.02,5.01,5.03,5.03,5.02,5.02,5.04,5.04,5.04,5.02,5.02,5.01,4.99,4.98
6,4.96,4.96,4.94,4.93,4.93,4.93,4.93,4.93,5.02,5.27,5.80,5.94,5.58,5.39,5.32,5.25,5.21,5.13,4
4.71,4.39,4.05,3.69,3.32,3.05,2.99,2.74,2.61,2.47,2.35,2.26,2.20,2.15,2.10,2.08
```

Training data

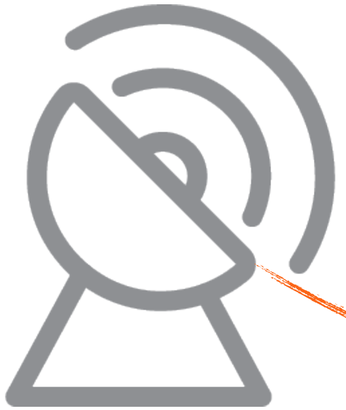
Training the model

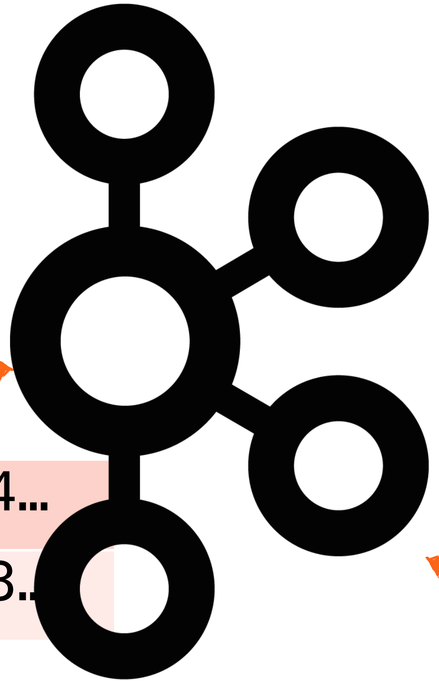
```
anomaly_model <- h2o.deeplearning(
x = names(train_ecg),
training_frame =  train_ecg,
activation = "Tanh",
autoencoder = TRUE,
hidden = c(50,20,50),
sparse = TRUE,
l1 = 1e-4,
epochs = 100)
```

```
2.10,2.13,2.19,2.28,2.44,2.62,2.80,3.04,3.36,3.69,3.97,4.24,4.53,4.80,5.02,5.21,5.40,5.57,5.7
79,5.86,5.92,5.98,6.02,6.06,6.08,6.14,6.18,6.22,6.27,6.32,6.35,6.38,6.45,6.49,6.53,6.57,6.64,
,6.73,6.78,6.83,6.88,6.92,6.94,6.98,7.01,7.03,7.05,7.06,7.07,7.08,7.06,7.04,7.03,6.99,6.94,6.
.83,6.77,6.69,6.60,6.53,6.45,6.36,6.27,6.19,6.11,6.03,5.94,5.88,5.81,5.75,5.68,5.62,5.61,5.54
9,5.45,5.42,5.38,5.34,5.31,5.30,5.29,5.26,5.23,5.23,5.22,5.20,5.19,5.18,5.19,5.17,5.15,5.14,5
5.16,5.15,5.15,5.15,5.14,5.14,5.14,5.15,5.14,5.14,5.13,5.15,5.15,5.15,5.14,5.16,5.15,5.15,5.1
14,5.15,5.15,5.14,5.13,5.14,5.14,5.11,5.12,5.12,5.12,5.09,5.09,5.09,5.10,5.08,5.08,5.08,5.08,
,5.05,5.06,5.07,5.05,5.03,5.03,5.04,5.03,5.01,5.01,5.02,5.01,5.01,5.00,5.00,5.02,5.01,4.98,5.
.00,5.00,4.99,5.00,5.01,5.02,5.01,5.03,5.03,5.02,5.02,5.04,5.04,5.04,5.02,5.02,5.01,4.99,4.98
6,4.96,4.96,4.94,4.93,4.93,4.93,4.93,4.93,5.02,5.27,5.80,5.94,5.58,5.39,5.32,5.25,5.21,5.13,4
4.71,4.39,4.05,3.69,3.32,3.05,2.99,2.74,2.61,2.47,2.35,2.26,2.20,2.15,2.10,2.08
```

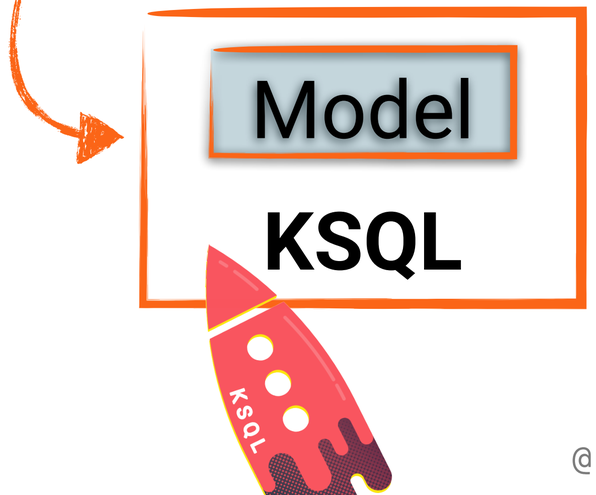


Applying the model to a stream of data





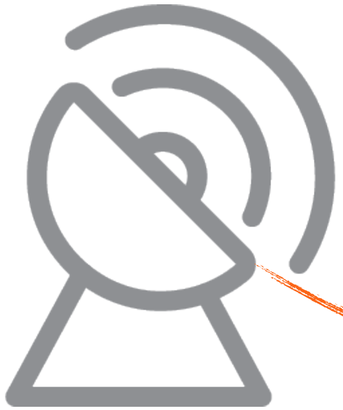
8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…

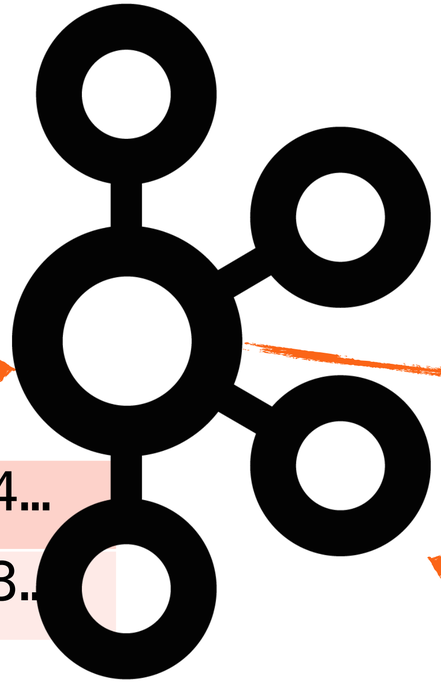


8	1.24412
9	2.12952

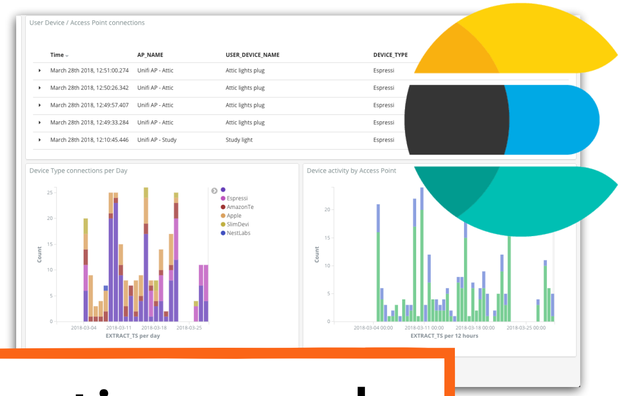
```
CREATE STREAM ecg_scored AS
SELECT event_id,
       ANOMALY(SENSORINPUT)
FROM health_sensor;
```

Stream the data to Elasticsearch with Kafka Connect





Kafka Connect



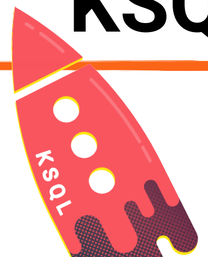
Elasticsearch

8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…

8	1.24412
9	2.12952

Model

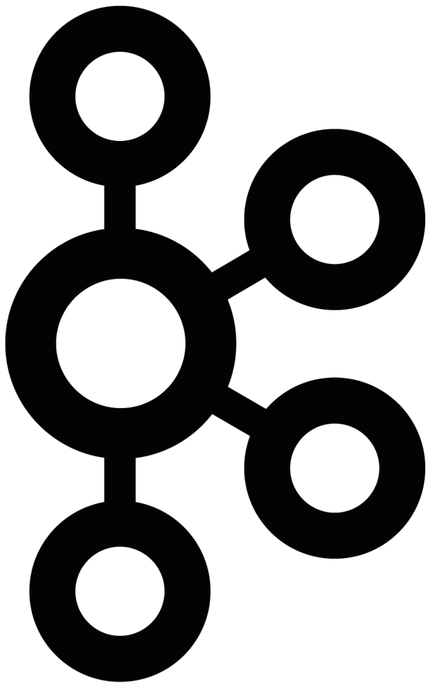
KSQL




```
CREATE STREAM ecg_scored AS
SELECT event_id,
       ANOMALY(SENSORINPUT)
FROM health_sensor;
```

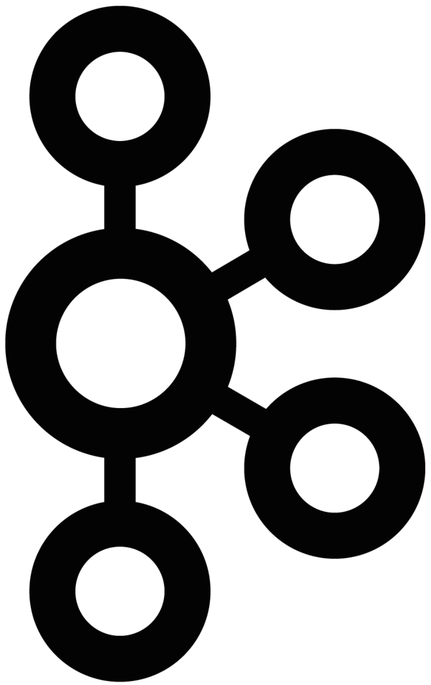
Event-driven, real-time Alerts on IoT data with KSQL

sensor_raw	
8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…



Event-driven, real-time Alerts on IoT data with KSQL

sensor_raw	
8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…

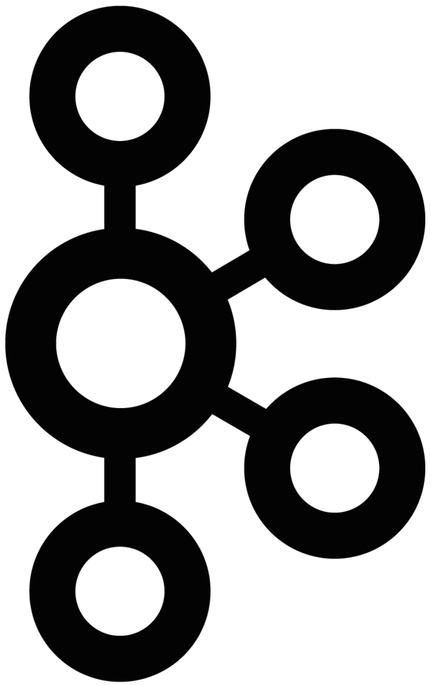


ecg_scored	
8	1.24412
9	2.12952

```
CREATE STREAM ecg_scored AS
SELECT event_id,
       ANOMALY(SENSORINPUT)
FROM health_sensor;
```

Event-driven, real-time Alerts on IoT data with KSQL

sensor_raw	
8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…



ecg_scored	
8	1.24412
9	2.12952

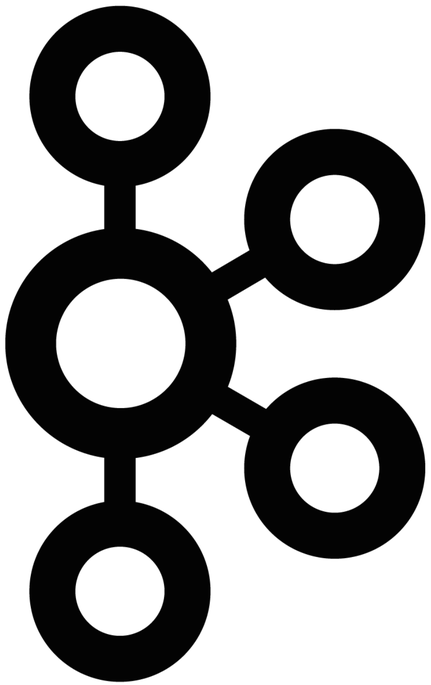
sensor_alerts	
9	2.12952

```
CREATE STREAM ecg_scored AS
SELECT event_id,
       ANOMALY(SENSORINPUT)
FROM health_sensor;
```

```
CREATE STREAM alerts AS
SELECT * FROM ecg_scored
WHERE ANOMALY > 1.5
```

Event-driven, real-time Alerts on IoT data with KSQL

sensor_raw	
8	6.00#6.04#6.08#6.10#6.14…
9	5.24#5.22#5.22#5.22#5.23…



```
CREATE STREAM ecg_scored AS
SELECT event_id,
       ANOMALY(SENSORINPUT)
FROM health_sensor;
```

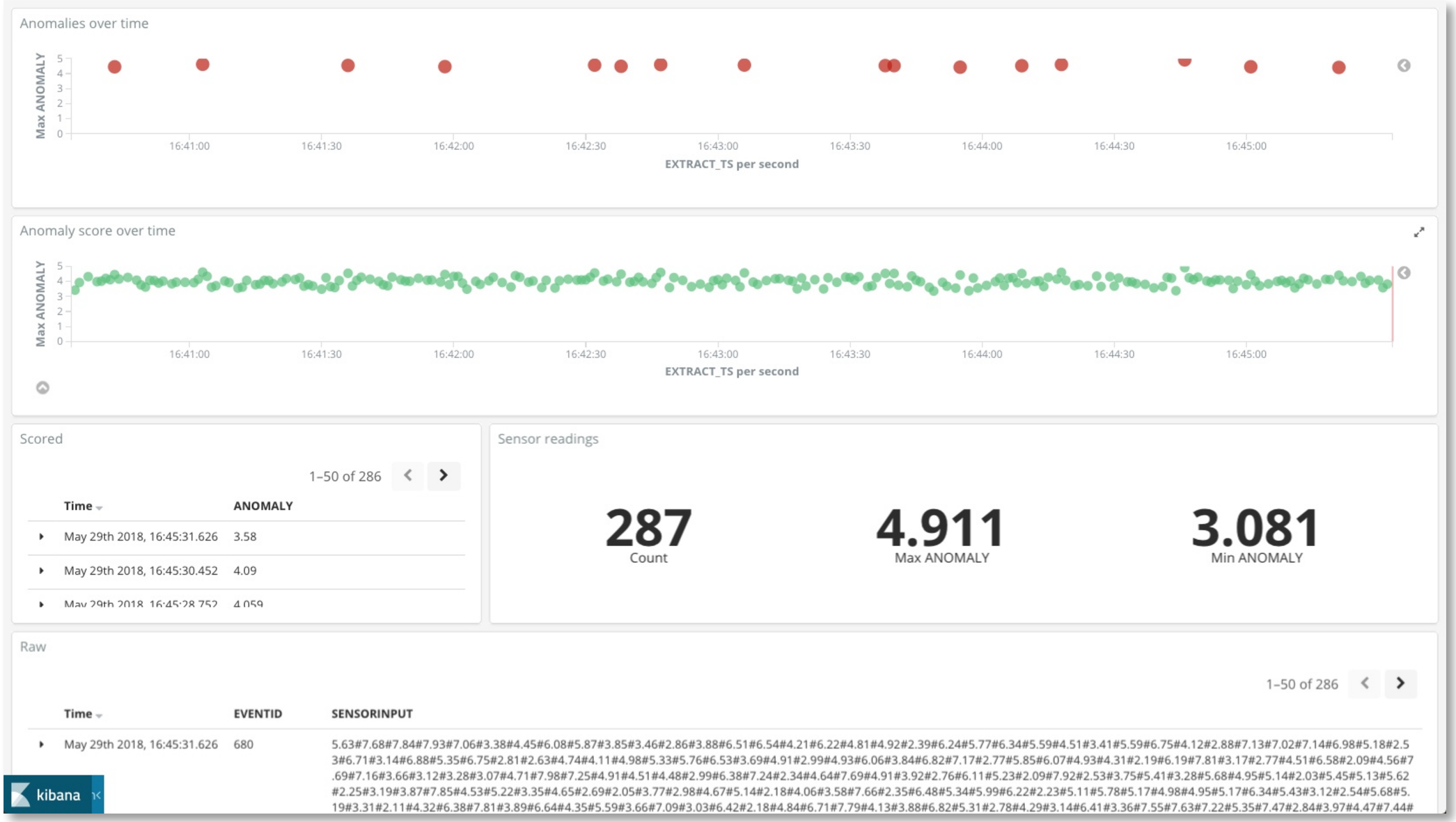
ecg_scored	
8	1.24412
9	2.12952

```
CREATE STREAM alerts AS
SELECT * FROM ecg_scored
WHERE ANOMALY > 1.5
```

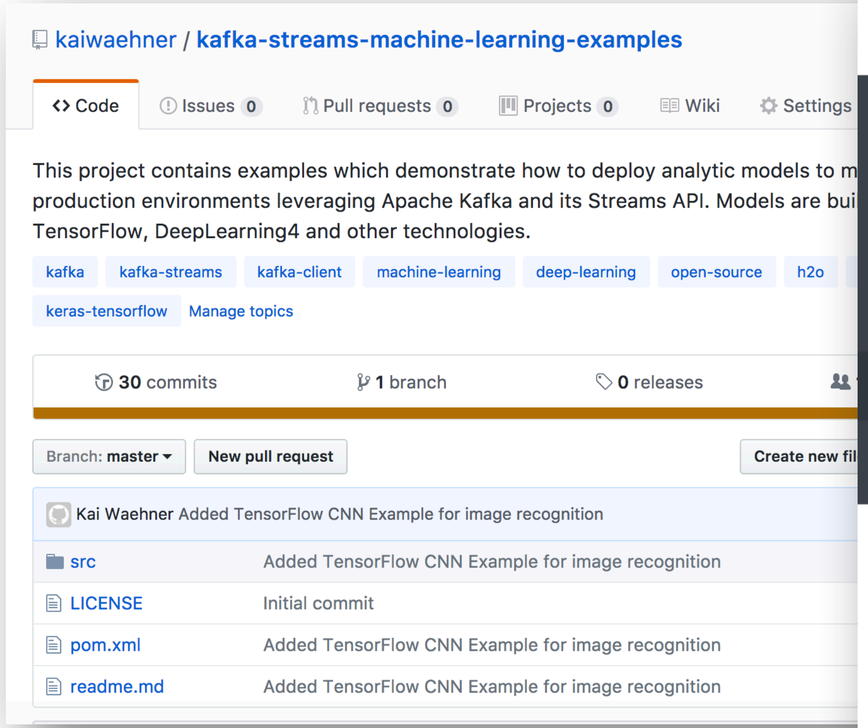
sensor_alerts	
9	2.12952

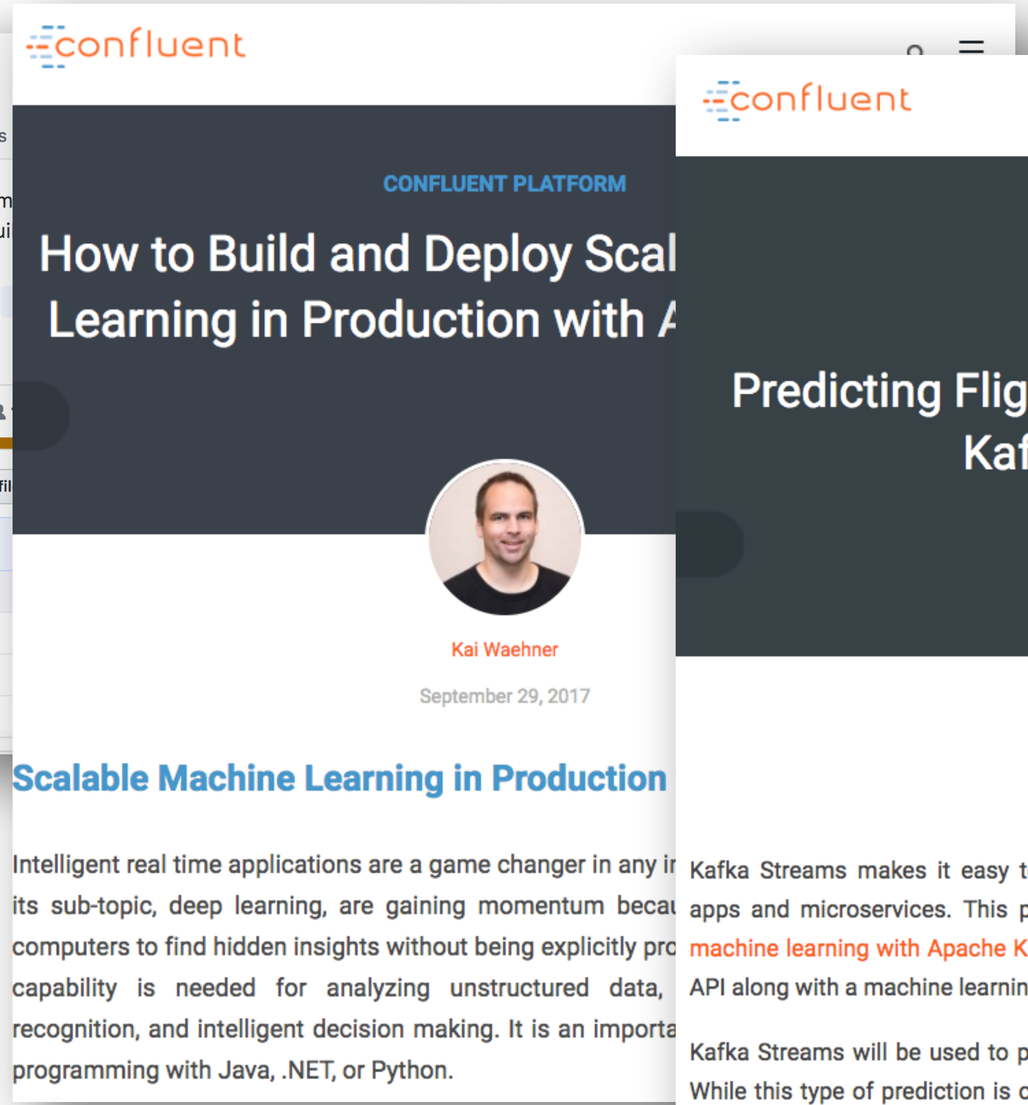
"AT 2018-05-31 14:55:09 BST EVENT ID 9 BREACHED THRESHOLD! DO SOMETHING!"

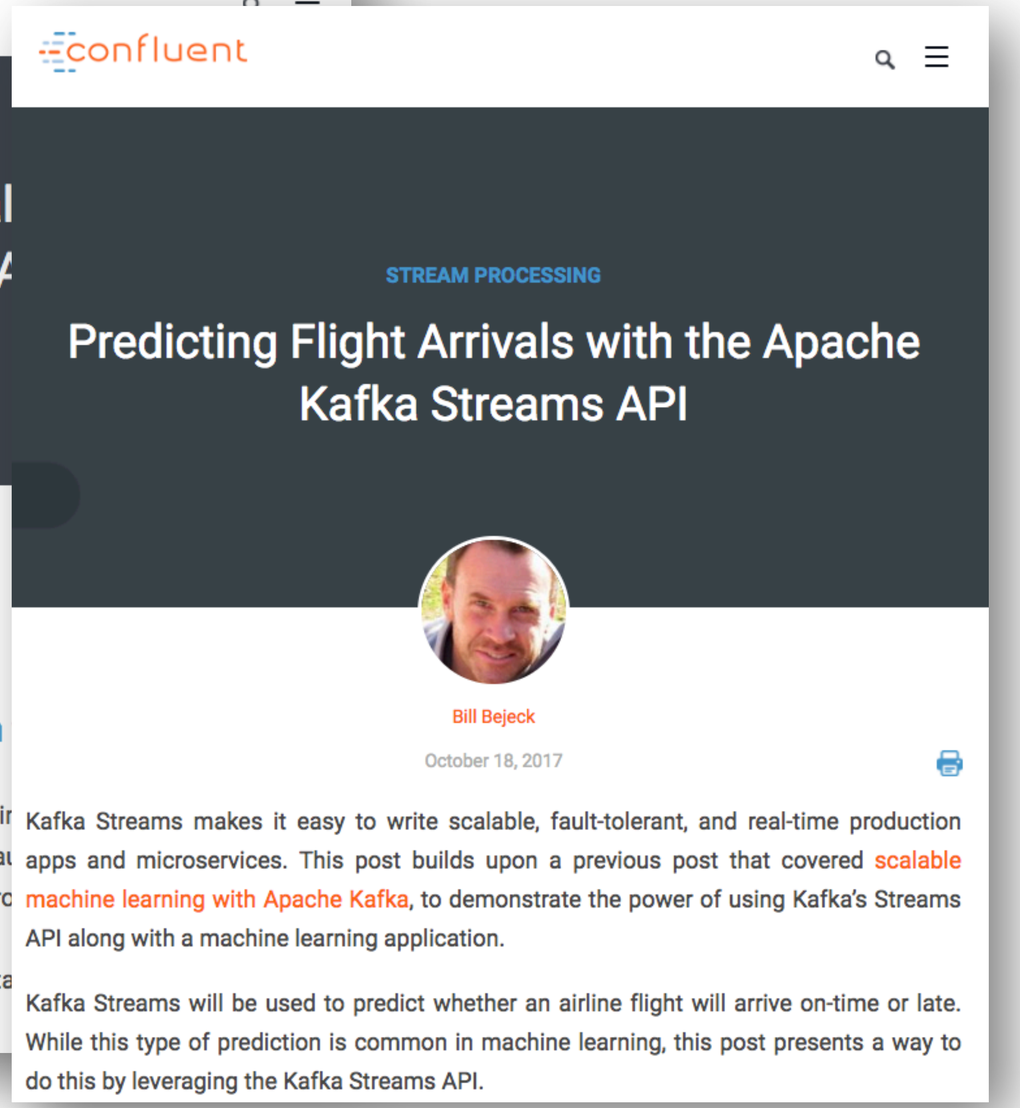
Visualise the data in realtime



Kafka + Deep Learning - resources



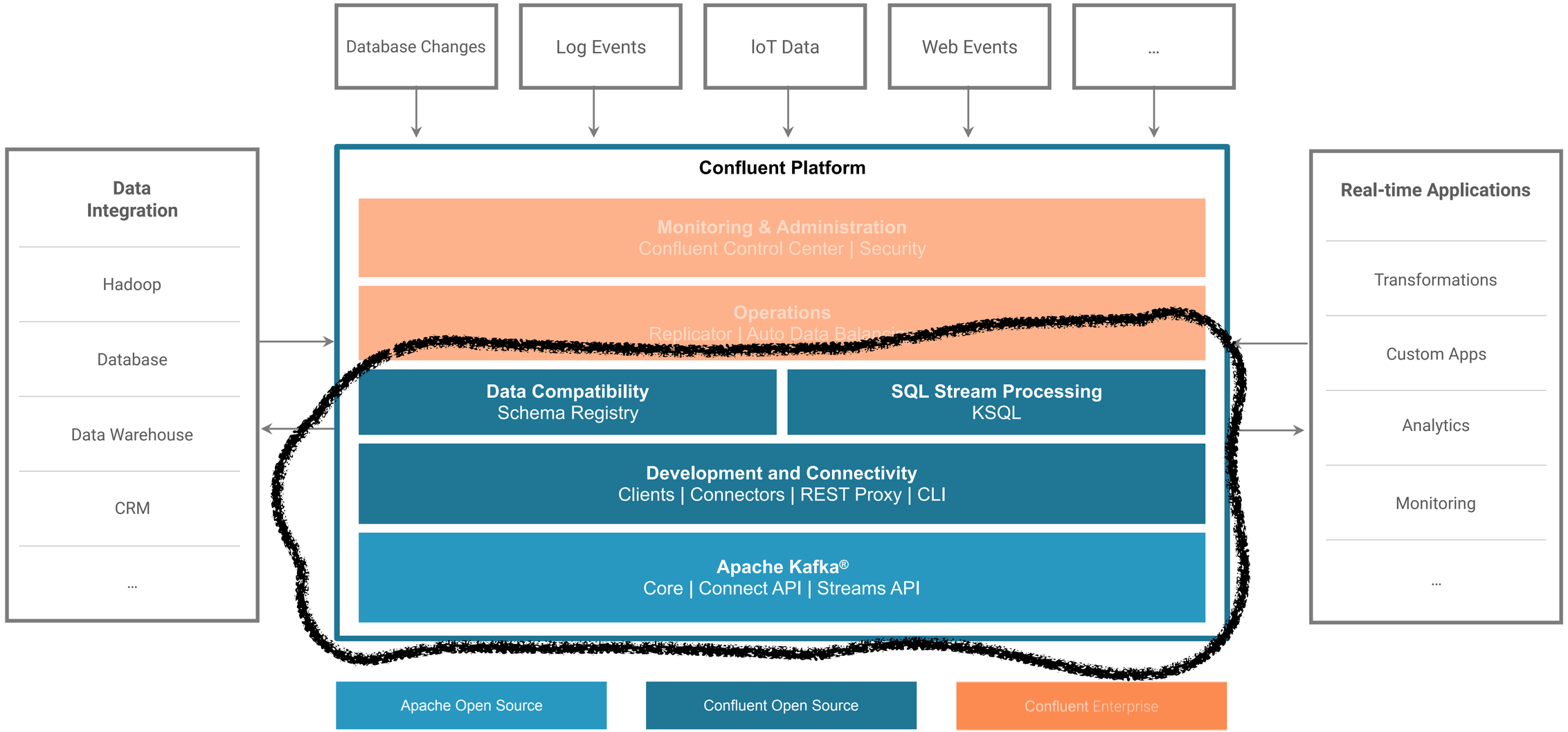




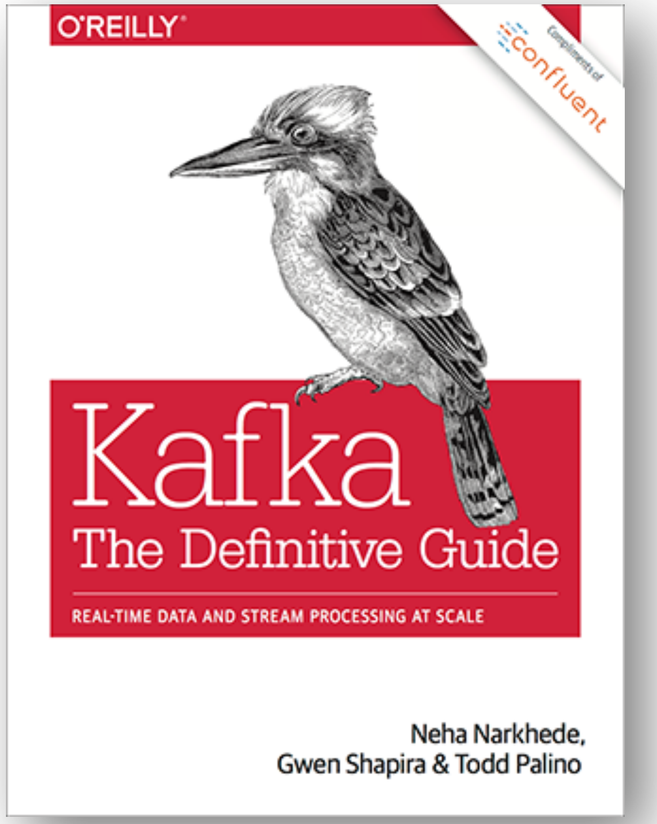
https://github.com/kaiwaehner

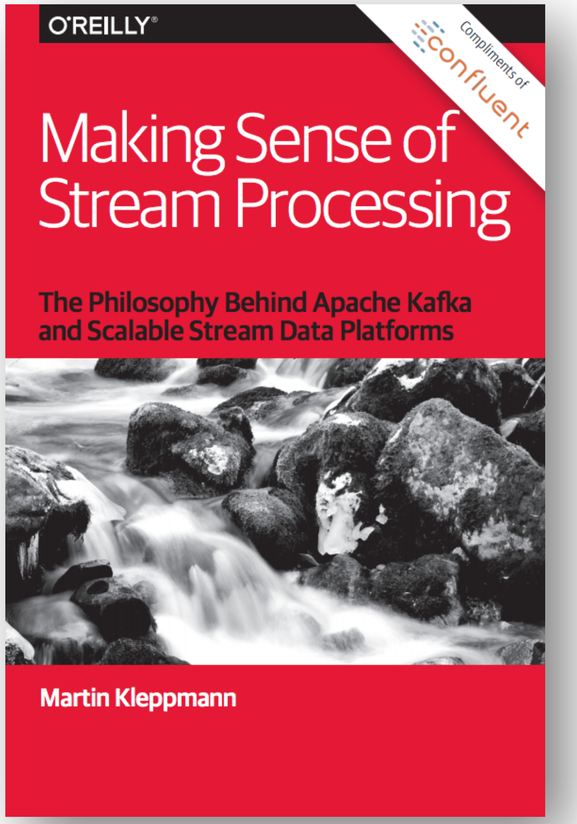
https://www.confluent.io/blog/

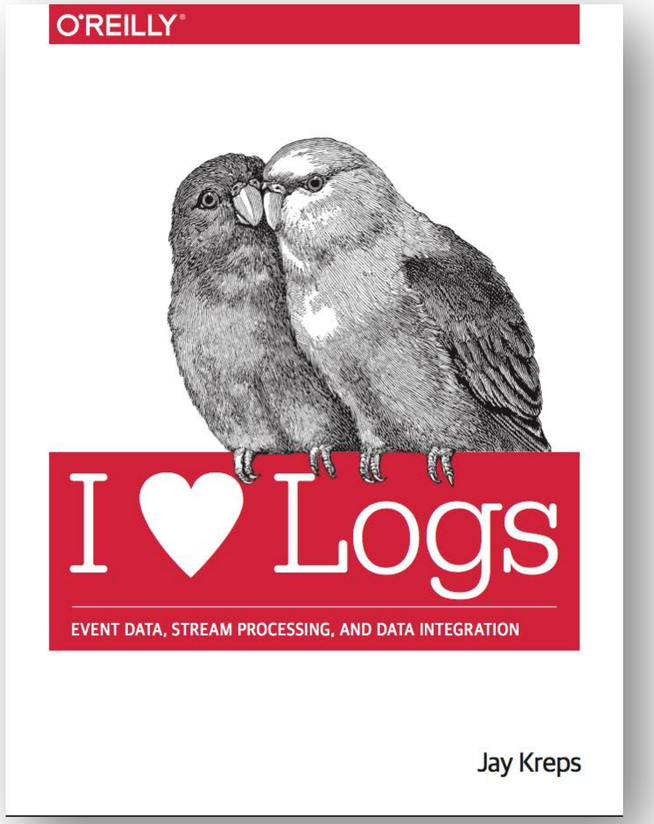
Confluent Open Source : Apache Kafka with a bunch of cool stuff! For free!



Free Books!







https://www.confluent.io/apache-kafka-stream-processing-book-bundle

https://www.confluent.io/download/
https://slackpass.io/confluentcommunity
@rmoff robin@confluent.io

#EOF